REPRESENTATION, STORAGE AND RETRIEVAL OF STORIES IN A SOCIAL SIMULATION

Robin Douglas Burke

Technical Report #50 • December 1993

NORTHWESTERN UNIVERSITY

Representation, Storage and Retrieval of Tutorial Stories in a Social Simulation

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Robin Douglas Burke

EVANSTON, ILLINOIS

December 1993

ABSTRACT

Representation, Storage and Retrieval of Tutorial Stories
in a Social Simulation

Robin Douglas Burke

Retrieving stories to present to students in a learning environment is a challenging and practical application of case retrieval. Opportunities to present stories arise at many points in the course of a student's activities and can take many forms. Stories can be used to provide a student with counter-examples, project the possible results of a student's choices, or critique those choices by analogy.

In this dissertation, I describe a retrieval system for tutorial stories, the Story Producer for InteractivE Learning or SPIEL. SPIEL is embedded in a larger teaching architecture known as GuSS (Guided Social Simulation). I describe the overall architecture of GuSS and SPIEL's place within it, using the example of YELLO, a GuSS application that teaches the fine points of selling Yellow Pages advertising.

SPIEL offers an alternative to traditional intelligent tutoring systems. It relies on its stories as a source of expertise about what it teaches, rather than an explicit expert model. Its pedagogical knowledge is in the form of a set of storytelling strategies, a library of different ways that stories can be used to teach. Each strategy entails a different kind of comparison between a story and a student's situation. These comparisons include assessments of similarity, dissimilarity and other kinds of relations.

SPIEL uses its strategies to develop procedures for recognizing storytelling opportunities. It examines each story in light of each storytelling strategy, arriving at a description of a tutorial opportunity: a situation in which the story could be told using the strategy. From these descriptions, the system builds rule-based recognition procedures capable of recognizing the opportunities when they arise in the GuSS simulation.

Acknowledgments

I've always wondered how Acknowledgments sections grow to such lengths, and now I think I know. The process of building knowledge thrives in the coming together of many minds and talents, and the work reported herein is no different.

Most important of these influences, of course, is my advisor Roger Schank. It will be clear to anyone familiar with his work that many of the ideas in this dissertation grew out of ones that he has long championed. Perhaps a less obvious but equally important influence he has had is through the creation of an exciting and challenging working environment: the Institute for the Learning Sciences. The fact that it continues to be a fascinating place in which to work is directly attributable to his vision.

After Roger, Alex Kass has been the most significant influence on my research, and I'm happy to say, a good friend as well. Alex was the first person I met when I arrived at Yale. I'll never forget that when we shared an office, we had to put a filing cabinet between us so we wouldn't converse so much. Alex got me involved in the ESS project (which turned into the GuSS project) at a time when I was floundering and helped guide my thinking about case-based reasoning and retrieval in more ways than I can count.

I felt Larry Birnbaum's influence even before I arrived in graduate school through Larry's lecture notes on natural language understanding. (Why he hasn't turned this stuff into the definitive text on the subject still escapes me.) Larry taught (and continues to teach) *the* introductory AI class. I have tried to channel some of his enthusiasm when my own reserve was failing.

My work on the GuSS project is what finally enabled me to crystallize my thinking about stories and teaching. I owe many thanks to the numerous members of the GuSS team, the most important of whom is Eli Blevis. Eli was directly responsible for the project from its earliest inception to its current realization and whose continuing work will result in more and better simulation tools not yet envisioned today. When we worked together, Eli's unique perspective forced me to question many things I would have otherwise taken for granted. He helped me analyze the problem of simulation monitoring and develop the basis for the current implementation of teaching modules in GuSS as simulation agents.

Smadar Kedar also brought an outside perspective to the GuSS project and SPIEL. When she started work on the Story Indexing Tool, it was a raw utility, something only I could use. She has brought it close to realization as a general tool for indexing. Our many conversations about stories, indexing, retrieval and rule generation have all contributed to clarifying my thinking on the issues in this dissertation.

Of the other GuSS hackers, I owe the most to Tom Murray and Greg Downey. Tom was there when all the central parts of GuSS were built and his insight into the system was invaluable for my understanding of how storytelling could be integrated. It was Tom who suggested the term "parasitic" to refer to rules that piggyback on the processing of simulation agents and he convinced me they were more than a hack. Greg turned my vague ideas about indexing into a reasonably slick tool, one that got me through the dark days of indexing. Keith Law spent two quarters figuring out what a MOP browser should do and then, amazingly, delivered it. Chip Cleary and David Foster also did some early hacking on the Story Indexing Tool.

For all of our tinkering, the GuSS project would never have gone anywhere without Michelle Saunders and Jeannemarie Sierant who wrestled with innumerable generations of tools and finally produced the world's first GuSS scenario. Michelle and Mary Williamson also were instrumental in the story-gathering interviews that resulted in YELLO's story base, and since that time Mary has contributed to our on-going efforts to build robust teaching modules for GuSS. She served as a guinea pig for early versions of the Story Indexing Tool and helped further my thinking on SPIEL's indexing scheme.

I must also mention other important contributors to the GuSS project: Jarret Knyal, Maria Leone, Charles Lewis, and Greg Sandell. SPIEL's video clips of stories would never have seen the light of day without the production efforts of Mark Schaefer and Max Miller.

YELLO was sponsored by Ameritech Publishing, and I am indebted to Diane Holcomb of API and Ameritech Institute fellows Mark Ashba and Debra Jenkins for their work both on YELLO itself and on building the bridges between organizations that were necessary for the project's functioning. John Miller, Joe Nussbaum, and Doug Wildey brought their expertise in sales and sales training to the Institute and ended up true collaborators in the design of YELLO. My research would not have been possible without the twenty-nine API account executives whose stories comprise YELLO's story base.

Earlier incarnations of SPIEL were in the domain of selling consulting services and supported by Andersen Consulting. For support and assistance with this phase of my work, I must also thank Marita Decker, Chris Sorenson, Maureen Zabloudil, Scott Rose and Suzanne Pink. For this application as well, I gathered a large story base, an endeavor made possible through the generous assistance of Leighton Smith and the Andersen Consulting managers and partners who contributed their stories while teaching at Andersen's Professional Selling School.

Many friends and colleagues at the Yale AI lab and ILS have contributed also to the ideas in this dissertation and to my general quality of life. I want to especially recognize Eric Domeshek and Chris Owens, whose research in indexing and retrieval strongly influenced my own, and Larry Hunter who was kind enough to show a fellow California boy the ropes. Thanks, Larry. Arman Magbouleh and Bill Ferguson were my collaborators on the Frequent Flyer program, where we hashed out many of the ideas about storytelling and conversation that are still under exploration at ILS today. I am grateful for the efforts of the many participants in Roger's indexing seminar in the fall of 1989 that produced the UIF technical report. Special mention goes to Heidi Levin for her many effort, not least of which was editing this manuscript before its publication as a technical report. Thanks also to my former officemate Mike Freed for our many discussions, Omega, contact juggling, and guerrilla Pictionary.

Before I ever arrived in graduate school, Bill Purves saw to it that I was bitten with the AI bug (more literally than you might guess), and pointed me in an interesting and challenging direction. Someday I'll get him back for that.

I have been truly fortunate to have the support of a wide circle of friends. Of special importance was my housemate Neil Carrick. Our adventures helped me make it through some tough Chicago winters. Dave Nicosia and William Kay were the source of much nutrition and entertainment. I hope to repay their generosity, but it just might be impossible. Many similar stories could be told, but I must forgo the details and just offer my thanks: Craig, Crystal, Dave (Billy-Bob), Dima, Gary, Gwenan, Jack, Jeanine, Kali, Karl and Rachel, Melodi, Patti, Sandrine and Kostya, Scott and Renee, Susan, Suzie, Veronique, Vicky and Eileen, and Ximena.

My parents have been a constant source of support of all kinds for me. Thanks for the infusions of California produce and local news. My sister, Victoria, was my first teacher; I think she must have been doing something right. I'll look forward to reading her dissertation soon. I also owe much to my large (and ever-expanding) collection of in-laws for their love and craziness, especially to Irene; Dick and Eve; Rebecca, Markus and Käthe; Jon and Mary; Betty and Hal; and Scott (PipeMan).

I reserve my final word of thanks for my wife Roshanna. She listened to and read countless rehashings of the ideas in this dissertation and indulged me through all of the psychic traumas that graduate students are heir to. Her enthusiasm, confidence, and love sustained me even when we were on different continents. I hope I can do as much for her.

I would also like to acknowledge some of the sources of financial support that have sustained me throughout my all-too-long student career, particularly the Holtville Rotary Club, and the General Electric Foundation. This work was supported in part by the Defense Advanced Research Projects Agency, monitored by the Office of Naval Research under contracts N00014-90-J-4117 and N00014-91-J4092. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. The Institute receives additional support from Ameritech and North West Water, Institute partners.

Table of Contents

List of Figures

List of Tables

List of Examples

To Roshanna

1. Introduction

1.1. Active memory and storytelling

The verb "remind" is deceptively passive. People "get reminded" as though it were something that happens to them without requiring effort or thought. Because the workings of memory are not easily accessible to introspection, it often seems that recollections arise unbidden to our minds.

The study of memory has often failed to question this passive bias. Psychologists frequently view memory access in passive terms: Anderson's (1985) discussion of association and activation is typical. The careful study of remindings, however, shows that they are governed by considerations of utility and function (Schank 1982). Simple association does not explain the influence of abstract, functional considerations in remindings (Seifert et al. 1986; Birnbaum and Collins 1989). An active, dynamic memory model (Schank 1982) is required, one in which recollections of particular types are actively searched for in memory, and one in which the purpose a retrieved item will serve is an important consideration (Kolodner 1984; Ram 1989; Hammond 1989).

One area where this model seems particularly applicable is the problem of retrieval for communication. Consider an ordinary conversational exchange:

A: Bush will have no trouble winning re-election. He just won a war.

B: I don't know about that. Churchill came out of World War II a winner, but he lost to Attlee in 1945.

In order to find an example to make a particular point in conversation, memory must be responsive to all the considerations that enter into the communication task. To find the Churchill example, B had to use knowledge of both politics and argumentation to identify what would count as a good counterexample to A's claim. Conversational storytelling, peripheral as it may seem, is therefore a good place to start in understanding how memory retrieves episodes (Schank 1990b).

1.2. Stories and teaching

This dissertation is about active memory for *stories*, episodes of human intentional behavior organized for communication, but it is also about teaching: about finding relevant stories and presenting them to students who are engaged in learning-by-doing.

For a simple example of how such a tutorial storyteller could work, suppose a student is playing the role of prime minister in a simulation of a parliamentary government. The student wins a major victory, such as negotiating a favorable trade pact, and considers calling an election. A storytelling tutor could see this as a good opportunity to present the story of Churchill's defeat as a warning to the student.

Tutor: You are considering calling an election since you've won an important victory and are doing well in the polls. You expect to do well, but consider this case where a similar expectation failed:

Churchill came out of World War II victorious and quite popular among the public. He called elections in 1945. It turned out that people were tired of the deprivations of the war, since rationing was still continuing in 1945, and Churchill's candidacy reminded them of those problems. Churchill's opponent, Clement Attlee, promised to build a new future for England in the post-war era. Churchill's personal popularity remained high, but Attlee became prime minister.

Through this tutorial intervention, the student learns about history and parliamentary politics in a relatively painless way. The case of Churchill's loss in 1945 has interest that it would not have on a textbook page because it relates directly to the student's goal of getting re-elected. A tutor with many such historical episodes could teach history in a concrete and comprehensible way by recalling particular episodes when they are relevant to what the student is doing.

This general approach to teaching is known as the *case-based teaching architecture* (Schank 1990a; Edelson 1993), in reference to the model of case-based reasoning from which it is derived. Case-based reasoning is a model of human reasoning in which recollections of prior experience are key (Riesbeck and Schank 1989; Kolodner 1993). The reasoner stores previously-solved problems, and when a new problem comes along, solves it by finding the most similar previous example and attempting to apply the same solution, adapting where necessary. The case-based teaching architecture is intended to mesh with the student's need to acquire relevant cases. Since new students in a domain will lack the experience needed to have a large case base, the goal of a case-based teacher is to use students' involvement in a learning environment as a means of introducing important cases that they should know.

For case-based teaching to work, the tutor has to be able to pinpoint those moments in an interaction when a student can profit from a story, and it must know what kind of story to present. For such reasoning, a storytelling tutor has to have three interrelated kinds of knowledge:

Relevance knowledge: Knowledge about what features of stories are important for making different kinds of points, and what features of a student's situation make these points good ones to make.

Indexing knowledge: An organized system of labels for stories, so that the tutor can find an appropriate example among the large number it possesses.

Recognition knowledge: An ability to monitor the learning environment, draw reasonable conclusions about what the student is doing, and relate its observations to the contents of the story base, so that relevant stories can be retrieved.

This dissertation talks about all three kinds of knowledge and their employment in a storytelling tutor, the Story Producer for InteractivE Learning or SPIEL.

1.3. SPIEL

SPIEL is a case-based teaching system that finds relevant stories through the recognition of storytelling opportunities. It is part of the Guided Social Simulation (GuSS) architecture for teaching social skills using a learning-by-doing environment. GuSS incorporates several different forms of tutorial guidance of which SPIEL's storytelling is one.

The GuSS architecture has been applied to create several applications that teach social skills.[1] In this dissertation, I use examples from the most advanced of these, YELLO, which teaches account executives at Ameritech Publishing the fine points of selling Yellow Pages advertising. In all, SPIEL has 178 stories about performing this task, told by account executives with considerable experience and success in this job.

GuSS is a learning-by-doing architecture for social skills; students' primary activity in a GuSS application is simulated social interaction. In YELLO, the student is called upon to develop and sell an advertising program through a series of sales calls. The student converses with the customer, attempts to gather information and tries to make a convincing argument for a

[1] In addition to YELLO, there are also GuSS applications for teaching novice consultants how to perform a business review (Kass and Blevis 1991; Blevis and Kass 1991), for teaching consultants how to sell consulting services (Kass et al. 1992), and for teaching project managers how to conduct a performance review.

proposed advertising program. Stories are retrieved in response to the student's actions and customer's reactions.

1.3.1. SPIEL's stories

SPIEL's stories are everyday anecdotes, first-person narratives about actual experiences. It is useful to distinguish these stories from other kinds of cases that a tutor might use, such as design examples, re-enactments or invented cases. First-person stories have properties that make them particularly useful for instruction (Schank 1990b; Witherell and Noddings 1991), especially for complex social skills (Kass et al. to appear):

Authenticity: the fact that such stories come directly from a person's real experience,

Detail: the tendency of such anecdotes to be vivid and detailed, and

Cultural content: the way in which personal stories reflect a person's beliefs and values.

Students learning a skill appreciate first-hand perspectives on what they are learning. It is one thing to hear an invented case that makes a point (such as a fable), but quite another to hear someone describe a real situation in which that point was brought home. The authenticity of stories drawn directly from personal experience adds to their impact. In the parliamentary simulation example above, the student would probably be interested to hear the historical account of Churchill's loss, but hearing the man himself describe his electoral defeat (perhaps from letters, memoirs or taped interviews) would be even more compelling. Such a story also serves as a good analogy for the student since the story is told from the perspective of someone in the same situation as the student's simulated role. This is part of the appeal of apprenticeship situations in which this type of story is common (Lave and Wenger 1991; Orr 1986).

Stories are rich in detail that would be lost in abridgment. Consider example 1.1, a story about a salesperson trying to sell Yellow Pages advertising. It shows a frustrating circumstance, a customer who put everything off until the last minute. The situation is turned into a successful sale by fast thinking on the part of the salesperson, who notices the man's prize sailfish.

One could easily condense this story into simple words of advice: pay attention to the buyer's physical environment, but such a summarization seems anemic compared to the original story. There is much detail in the sailfish story that could be important to someone who is learning to sell Yellow Pages advertising. For instance, a novice would not necessarily know about the strict deadline for closing accounts in a Yellow Pages book and the impact it has for selling, a point that the story illustrates quite well.

The story also shows how a difficult customer can be disarmed. Novices are frequently misled by a customer's initial reaction: a beginner might look at the buyer's question, "Do I have to sign something?" as an opportunity to get a difficult sale closed and over with. The change in the customer's demeanor after talking about his prized catch might be unexpected to a beginner, but an experienced salesperson knows that customers frequently put themselves "on guard" when a salesperson is around and can be brought into a more receptive frame of mind with well-chosen conversation.

These details make stories told from real experience vivid and memorable. Students learn many things about the world of the practitioner beyond the story's plot or point. Indeed, the value of detail in experientially-based knowledge is one of the important claims of the case-based reasoning model. If a case is simplified into a single abstract statement such as a rule, the reasoner loses access to any of the other lessons that could be gleaned from it. Since it is difficult to anticipate all the ways that a case can be used, a case-based reasoning system keeps all the details of the case around.

I was in Muncie, Indiana. I had tried and tried to reach this guy. He was a grocer. He had a large ad under groceries and a large ad under bakeries. I had never called on him. All through the canvass, I could not reach him. He was busy, unavailable. It got down to the last day or the next to last day of the canvass, and she said "Yeah. Come on out. He'll see you."

I went out. I sat in the waiting room of his office for an hour. I was really upset. I walked in. The guy was sitting at his desk. Behind him, in this little office, was a sailfish that stretched the length of the wall. It was about a seven-foot sailfish. Beautiful.

Anyway, as I walked in, he was real busy. He was doing payroll. He looked up at me and said "Do I have to sign something?" Seems like people are always asking that. My answer to that was "Did you catch that sailfish?" He looked up at the fish. He looked back and he said, "Why, yes I did." I said "Where did you catch it?" He proceeded to tell me where in Florida, down in the Keys, that he had caught the fish. He talked for at least 15-20 minutes about that trip, about five years ago it was. And he talked and he talked. I think all I did was say "Wow," "Gee," "No kidding," "I'll be," "I don't believe that."

Twenty minutes later, he looks at me. This is a guy that wouldn't give me the time of day, wouldn't even return a call for five weeks. He says "Do you have something to show me this year? You got something new?" I laid down a couple of pieces of spec. And he said "Oh, I like this." It was one of the first years that we had color. He said "I like the color." He asked how much it was. I told him. He said "That's fine. Let's go with these."

I made one of the best sales in Muncie, all because I asked if he caught the sailfish. What I did there was I took advantage of something that that man was very proud of. He would not have had that on the wall if he didn't want someone to say something. The fact is it would have been rude of me not to mention his sailfish. I didn't dream he was going to go into a 20-minute dissertation, but it can be anything physical in that office that that man is displaying. He's obviously proud of it. Recognize it. Let him know. If they don't want to take off on it, that's fine, but this guy did. He bought from me simply because I noticed his sailfish.

Example 1.1. The "Did you catch that sailfish?" story[2]

The demands of authenticity and detail encourage the use of the most vivid means of story presentation. One effective technique is to record on video the act of storytelling and replay it for the student. Research in video-based learning environments has discovered that students find stories of this type quite compelling when presented in this way (Ferguson et al. 1992; Slator et al. 1991). Even stories that are fairly lengthy such as the above example can maintain interest when presented on video at the right time. Using optical disk or digital video, stories recorded in this manner can be stored in large numbers and easily displayed for the student.

Re-enactments can also have the properties of authenticity and detail. However, stories have another important feature in that they are cultural artifacts: they are created by people in order to communicate their experiences to others (Polanyi 1985). When experts tell their stories about selling Yellow Pages advertising, they teach something of their *culture of practice* (Lave and Wenger 1991). Because stories include not just a recounting of events but also a person's perspective on those events, students learn not just what happened but how events were thought about and perceived by participants. This is particularly important when students are trying to become members of a community, such as the community of Yellow Pages account executives: Lave and Wenger note that learning the stories of a culture is an important way that a novice can begin to participate in it.

Although stories are crucial for learning, by themselves they are a poor curriculum. One could construct a documentary film about selling Yellow Pages ads by stringing together stories, but

---

[2]This story and all the other examples used in this dissertation are transcriptions made from videotaped interviews with experienced account executives at Ameritech Publishing in Troy, Michigan, on July 14 and 15, 1992. Interviews were conducted by Alex Kass, Michelle Saunders, Mary Williamson and myself. This story was told by Denny Gant and was transcribed from video clip #254. Titles were added during transcription.

it would be difficult to make such a film interesting, even for a committed learner. It is hard to understand and make sense of the point of a story without having a context within which to place it.[3] For this reason, the case-based teaching architecture requires that students be actively participating in an engaging task so that cases will be most effective.

1.3.2. An example of SPIEL at work

The task of students using the YELLO program is to try to sell Yellow Pages advertising. SPIEL watches their actions and the customer's reactions and looks for opportunities to present relevant stories. Consider the following interaction:

Student: So, we're going to go ahead with the 1/4 page ad with color?

Customer: OK.

Student: Just sign right here.

Customer: [signs]

Student: I think the color is really going to attract people to your ad.

Customer: I sure like the way it looks.

Student: Ask your customers what they think. I'll bet you'll find it's an attention-getter. Thanks for your time.

Customer: See you next year.

Student: See you. [leaves]

This is a successful sales call, but SPIEL has a cautionary story to tell. Every moment that the student remains after the close of the sale gives the customer an opportunity to retract his buying decision. Although that did not happen in this case, it is a possibility that the student should be aware of.

SPIEL intervenes in the following way. First, it signals to the student that it has a story available:

SPIEL: A story showing the risks of your approach.

Then, if the student expresses an interest in hearing the story (through a button press), it explains why the story has come up:

SPIEL: You kept talking to the client after the sale was closed. Nothing bad happened but here's a story in which doing that led to problems:

Then the tutor shows a video in which an experienced salesperson relates a personal experience.

Video: I was in the South Bend/Mishawaka area. This was my first or second year. I was calling on a swimming pool contractor. He had quarter page in South Bend. I was proposing a quarter page in another directory. It was sitting at his kitchen table. And the guy was hesitating; he didn't know... So, after a few more attempts, he says to me "OK, I'll go with the other quarter page." He bought it. I pushed the order form in front of him. He signed it. It's done.

As I'm putting my stuff together, I made this comment that cost me the quarter page. I said, as I'm packing up, "I'm sure that you're going to get a lot of phone calls from that new ad." He looked at me and he said, "You know, I don't need any more phone calls. I'll tell you what, let's not do that this year, maybe next." I talked myself out of

[3] One example of the successful use of a stories-only documentary can be found at the U.S. Holocaust Memorial Museum in Washington D.C. A continuous display of first-person stories is shown, but only at the end of the museum tour, when visitors have absorbed the context that the exhibits vividly provide.

> a quarter page. I've never done it since. I walked out. There was nothing I could say. I had it and I lost it. All I had to say was "Thank you very much Joe. See you next year." But I didn't. I had to tell him about the calls, which I'd already done twenty times.[4]

Finally, the storyteller sums up the story for the student:

> SPIEL: Nothing bad happened to you because you kept talking to the client after the sale was closed, but sometimes the client changes his mind.

In this example, the storyteller augments the student's simulated experience in an important way. Without the "Talked myself out of a sale" story, the student, who was successful, might never realize the risks inherent in remaining after the sale. The story arrives just at the time when it is most relevant, after the risk is past and the student thinks all went well, and it is exactly on point as a counterexample: it shows a situation in which the same tactic had an opposite outcome.

For SPIEL to intervene in this way, it needs the three kinds of knowledge discussed above. It has to recognize that a story about a failure to sell is relevant when the student has succeeded in selling using a similar tactic. It has to have a library of kinds of **relevance** that includes such notions as showing stories with opposite outcomes. In order to know that the "Talked myself out of a sale" story has relevance in this situation, SPIEL needs **indexing** knowledge. Its representation of what the story is about needs to include concepts such as the plan of continuing conversation after the sale is closed and the outcome of having the customer change his mind. Finally, in order to see that what the student is doing constitutes a storytelling opportunity, SPIEL has to have knowledge of **recognition**. It has to know that only after the student leaves has the danger passed that the customer will have a change of heart, and that the customer's signature constitutes the completion of the close of the sale, among many other things.

1.4. Relevance knowledge

In asking the question "What makes a story relevant?" I am deliberately taking a different tack from that typically found in computer-based learning environments, such as intelligent tutoring systems (Polson and Richardson 1988). Such systems are usually extensions of today's formal schooling. The designer of the system develops a set of instructional goals, puts together a computer environment where the knowledge relating to those goals can be transmitted and tested, and develops presentation material that helps students meet the instructional goals.

This is a less appropriate design for a learning environment for teaching social skills. There are usually no hard-and-fast rules about the exercise of social skills, and often experts will have widely-differing opinions about what exactly a student needs to learn. Many complex skills have the property that it is difficult to state exactly what is required for mastery (Funke 1991).

A case-based view of expertise contributes a different notion of curriculum. To be good case-based reasoners, students need to learn a variety of cases so they will have background knowledge from which to reason when confronting new problems. Every one of SPIEL's stories is a real-world case that an expert found important enough to save and recall. What better source of a curriculum than the cases on which real expertise has been built?

Using stories as the primary pedagogical source entails a different notion of tutorial intervention than found in intelligent tutoring systems. The relevance of a story for a

---

[4]"Talked myself out of a sale," story from interview with Denny Gant, video clip #150.

storytelling tutor is not based on a notion of the correct or incorrect way to perform a task. What characterizes a storytelling opportunity is the existence of a story that will catch a student's interest sufficiently well that the story will be understood and remembered. Rather than modeling what the student knows about the task and trying to make sure the student meets an expert standard, a case-based teacher has the easier task of representing what the student is likely to be interested in (Edelson 1993).

Stories can be intrinsically interesting for many different reasons – they can be funny, grotesque, dramatic, mysterious, etc. – but SPIEL is not interested in stories for their own sake, it does not strive to be a conversationalist. As a tutor, its role is to help the student, implementing what Papert (1980) calls the "power principle," the notion that a tutor should enable students to do something that they want to do. SPIEL's stories must therefore be interesting because they are relevant to the learning that students are already performing in GuSS's learning environment.

1.4.1. Failure-driven learning

Learning-by-doing is frequently driven by failures: unsuccessful results or unmet expectations. Failure-driven learning theory (Schank 1982) holds that there is a central cycle in this type of learning. The learner makes predictions about the results of its own actions or the behavior of others, and tests these predictions by observing what happens. If all goes well, these expectations feed into decisions about the next action or prediction. However, if the expectations fail, the learner has made a mistake in understanding the world. The failure helps focus attention on exactly what may be wrong in the learner's model of the world or in its processing. To prevent future failures, the learner must explain how the failure occurred in sufficient detail to allow the problem to be fixed.

The failure-driven learning cycle makes it possible for a student to recover from errors independently of instruction. A student who hangs around after making a sale and has the customer change his mind may come to understand the importance of leaving after the close. However, such learning is dependent on the student having the right kinds of experiences, and requires that the student be able to assign blame to the right error, which can be a difficult explanation task.

The ability of a student to engage in solitary failure-driven learning is therefore dependent on the student's knowledge of the domain. In a domain where the student knows very little, failure-driven learning will degenerate into unproductive thrashing around, since the student cannot perform any part of the learning cycle well. Unassisted failure-driven learning works best when the student knows enough about the domain to make the right kinds of observations and to construct reasonable explanations. When this is not the case, a tutor can assist by giving students the knowledge that they need to form good explanations.

Even knowledgeable students, however, will not always be able to learn because the variability and complexity of the real world means that they will not necessarily encounter good learning experiences. If the student gets away with using a risky tactic, as in the example, a story can make the risk obvious by showing an example of a bad outcome following from the same tactic. It can spur learning by getting the student to recognize a possible problem.

There are therefore two sources of relevance for stories in a learning-by-doing environment: (1) they can point out errors that the student may be making when the environment does not provide sufficient feedback for the student to notice them, and (2) they can help the student explain expectation failures when the environment does not give all the information needed to construct an explanation.

1.4.2. SPIEL's storytelling strategies

To get its cases across, a case-based teaching system needs an understanding of where the student's interests lie. In a learning-by-doing environment, interest will be governed by relationships between what the tutor presents and the issues of failure-driven learning: expectation failure and explanation. Building a system capable of recalling relevant stories entails explicitly representing these relevance relationships.

SPIEL's relevance knowledge comes in the form of *storytelling strategies* that represent the conditions under which a story with particular characteristics will be relevant to a student in the GuSS environment. In the example above, the "Talked myself out of a sale" story was told using the **Demonstrate risks** strategy. This strategy selects stories about the failure of a particular course of action and tells them when the student has executed a similar course of action but had success. The story is relevant because it contradicts the evidence of the simulation with respect to the student's evaluation of his or her recent success. This is an educationally-appropriate way to intervene since, without the story, the student might not recognize the possible risks the story demonstrates.

SPIEL has thirteen such storytelling strategies. Each has a similar function: selecting stories of a particular type and characterizing situations in which they will be relevant. The strategies fall into four categories based on their relationship between the stories they present and the situations in which they present them:

Strategies that show the student alternatives. These strategies present stories whose outcomes differ substantially from what the student has achieved in the simulated world. **Demonstrate risks** falls in this category. Confronting students with stories about alternative possibilities gets them to question their expectations about the simulated world. Also in this category are the strategies **Demonstrate opportunities** and **Demonstrate alternative plan**.

Strategies that critique the student's expectations. By showing examples of situations where people in a similar role had preconceptions that were incorrect, these strategies help students transfer expectations from their everyday social lives to the specialized social environment they are learning about. There are six strategies in this category, three basic types, each of which has 2 sub-types. The basic types are **Warn about optimism**, **Warn about pessimism**, and **Warn about assumption**.

Strategies that project possible results of actions the student is taking. Normally, students receive immediate feedback from the execution of their plans. If they do not get such feedback, projection strategies call for the storyteller to provide examples that show possible outcomes. There are two strategies here: **Reinforce plan** and **Warn about plan**.

Strategies that explain the perspectives of other people to the student. These strategies recall stories that explain why people acted as they did in real world situations, making the unexpected actions of others in the simulation comprehensible. In this category, there are two strategies: **Explain other's plan** and **Explain other's perspective**.

These categories cover the most important ways that stories can be relevant to students using GuSS. They concentrate especially on plans and outcomes, which is consonant with the emphasis in GuSS on learning-by-doing. Students are largely engaged in planning and acting in the social environment. In a different kind of task, such as a design task, students would focus on different kinds of expectations, such as how a design feature will achieve a particular function and how it will interact with other features. Different kinds of strategies would be needed to address these expectations.

A good example of how storytelling strategies are different for different learning environments can be seen in the case-based teaching system Creanimate (Edelson 1993). Creanimate is a case-based teaching system for elementary biology, specifically in the area of anatomical form and function. Its learning environment is fanciful: students attempt to design their own animal by modifying one that already exists. The system then carries on a dialogue intended to address the issues of form and function that arise due to this modification.

Students' interests in Creanimate focus on the issues of biological form and function, so Creanimate uses case retrieval strategies that are organized around these principles. For example, the student might want to design a butterfly that fights. The system uses the student's interest in fighting as a springboard to cases about other animals that fight and the reasons why they do so.

There is not a lot of emphasis on explanatory stories in SPIEL's storytelling strategies, because students in GuSS already have a great deal of social knowledge. The one area where students do get help is in the explanation of the unexpected actions of the characters in the simulation, since students may be unfamiliar with how people behave within the specialized social sphere they are learning about. In a domain where students cannot be expected to have a lot of knowledge, explanatory stories become much more important. Creanimate's students are not familiar with the particulars of animal adaptation, so it concentrates on using stories to explain. This emphasis means that its strategies look for very simple connections between the student's design and the organization of real animals, so that the explanatory relationship is clear. More complex strategies would be needed if the system were to teach about the complexities of the design process itself.

SPIEL uses its library of storytelling strategies to compare a story and a student's situation and determine if the story is relevant. In order for this comparison to be possible, SPIEL must know what its stories are about. Stories must be labeled so that the features crucial for comparison are readily available, and the system of labels must be organized so that the retriever can avoid having to look at every story.

1.5. Indexing knowledge

The problem of designing a representation to be used for labeling cases is known as "the indexing problem" (Schank 1982; Domeshek 1992). An index has to include features that are good predictors of a case's usefulness. For example, the fact that the customer in example 1.1 originally has the intention of minimizing the time spent with the salesperson is more important than the fact that it is a sailfish on the wall instead of a diploma. The story will be worth telling when the student encounters other customers who claim they have no time for talk, whether or not they fish.

One proposal for tutorial cases is to index them based on the situations in which they should be retrieved (Edelson 1993). For example, the "Talked myself out of a sale" story could be given an index like "Tell this story when the student makes a sale, continues talking to the client, but doesn't lose the sale." The problem with this approach is that it requires that the creator of the index consider all possible uses at the time that the case is indexed.[5] It means that there will be a new index for every way that a case could be used.

The "Talked myself out of a sale" story was used with the **Demonstrate risks** strategy in the example, but it could also be used with the **Warn about plan** strategy. For this to happen, the bad outcome, the customer changing his mind, would have to be somehow delayed in time. For example, the student could make a sale and then suggest that the customer come out to

---

[5]Automatic indexing of video material is not feasible. Indices for SPIEL are created manually.

lunch with him. **Warn about plan** suggests that the story should be told at this time to make essentially the same point as before: that hanging around after the sale is dangerous. Using this strategy, the story would be told before the lunch happens so that the student has a chance to avoid this dangerous situation. This use of the story entails a different set of retrieval conditions, and therefore the story would have a different index for each strategy it could serve, if SPIEL indexed stories by their retrieval conditions.

SPIEL's alternative is to use an indexing representation that captures properties of the story itself, independent of possible tutoring function. The only consideration at index creation time is what the story means, not how it ultimately may be useful. With this indexing scheme, the index for a story need only be written once. The storytelling strategies examine each index in order to decide how and when to tell the story.

1.5.1. Anomalies and explanations

Stories frequently encode interesting expectation failures, or *anomalies* (Schank 1982; Schank et al. 1990). In the "Talked myself out a sale" story, the salesperson clearly expected that his comment would reassure the client about the value of the ad he was buying. This hope was not realized: the client instead reconsidered the value of the ad and decided not to buy. This is the central problem, or anomaly, in the story.

Anomalies are what make a story memorable both to the teller and the hearer. One can readily imagine the storyteller's distress at having his reassuring intentions backfire. The teller experienced the original expectation failure, and developed the story as a case that encapsulated the experience. Stories about anomalies and their explanations address issues important for students' active learning: experiences of expectation failures and attempts to explain them.

The general form of an anomaly can be stated as follows:

> Actor X had an expectation that Y would happen, but actually Z happened.

A statement of the anomaly from the "Talked myself out of a sale" story is

> The salesperson expected the client would be reassured about his purchase but actually the client decided not to buy.

This anomaly captures one important aspect of what the story is about. A story usually says (or implies) more than this, including why the expectation came about and why it failed. The salesperson obviously thought that talking about the increased number of calls the client would get would be reassuring, and thought that reassuring the client was an important goal. The expectation failed because the time spent after the sale gave the customer time to reconsider.

1.6. Recognition knowledge

Indices represent what a story is about. Strategies represent how, in general, stories can be made relevant to a student. There is one final link that a storytelling tutor must make, which is the connection between students' actions and its notion of a relevant story. As shown in the "Talked myself out of a sale" example, this involves a variety of considerations including social knowledge, such as the scope of a conversation, and practical details such as when a sale should be considered closed. This recognition knowledge has three different types: (1) task knowledge, both general and specific, (2) student stereotype knowledge, and (3) manifestation knowledge.

Of primary importance for a tutor is a general understanding of the task. The tutor has to know that the student is trying to sell, and that selling involves sub-goals like gathering information, constructing presentations, making sales pitches, and answering objections. At a more concrete level, SPIEL has to know particular details of how the selling task is achieved by the student in the YELLO application. It has to know which kinds of answers from a customer constitute substantive information and which are evasions, what actions constitute the construction of a prospective ad, and so forth. The task of representing these details is of course greatly simplified by the fact that SPIEL is operating in a simulated environment with limited scope. It does not have to be concerned about every possible way a student can ask a question; it need only recognize those choices that GuSS actually permits.

SPIEL's knowledge of the student takes a much simpler form than the student models found in intelligent tutoring systems (vanLehn 1988). As I have discussed, SPIEL needs to represent students' states of interest, not their states of knowledge. It is sufficient for SPIEL to have a static notion of what will interest students since the things that they will find interesting do not change as fast as the things that they know. For example, in order to use the strategies that explain other's behavior, SPIEL has to know what actions students will find unexpected. It has a static set of stereotypes that students are assumed to possess and tells stories when they show customers and others doing things that violate the stereotype.

SPIEL's expectation-directed strategies require that the system be able to tell stories that contrast with expectations that the student is likely to have. Of course, SPIEL cannot read the student's mind. There will always be the possibility that the student has a certain assumption, but takes no action that clearly indicates it. There is little that any tutor can do about this, short of asking students directly what they are thinking. What SPIEL can do is identify situations in which there is a clear manifestation of a student's expectations. If a student constructs a large ad campaign as part of a sales presentation, this is pretty good evidence that the student expects the customer to be willing to spend a lot of money. SPIEL's manifestation knowledge allows it to infer what symptoms it can observe that are likely consequences of a student having a particular expectation.

1.7. Retrieval architecture

Even given all the appropriate knowledge, it is still no small matter to locate a relevant story out of a library of many. The way a retrieval problem is usually posed is in terms of cue formulation. There is one process that decides what type of information is needed and creates a retrieval cue, a description of what should be retrieved. A second process searches memory for a good match to this description.

Cue formulation is not the best way to think about tutorial storytelling. If there is no story to tell about the student's situation, the work invested in recognizing the need for a story and formulating a retrieval cue is wasted. In SPIEL, a cue formulation process would have to derive features like "the student has an assumption that is likely to fail" in order to search for a story with which to use to critique the student's assumptions. How can the tutor know what assumptions of the student's are likely to fail? Indeed, given that recognizing any aspect of the student's knowledge is difficult, how can the tutor know what assumptions it should try to recognize? Reasoning from first principles is not likely to shed much light on these questions, given that social knowledge is both vast and idiosyncratic.

A better approach is to use the stories themselves as source material. SPIEL's stories are, in effect, a case-based model of a social task. To critique a student's expectations, SPIEL need not develop a model of the student's thinking and pinpoint those expectations that may be suspect. Its stories tell it what expectations other salespeople incorrectly made. If a story talks about a failed assumption, SPIEL knows that at least one person had this assumption and had it violated. If it can identify that the student may have a similar assumption, then the story will be relevant. Instead of diagnosing the student's failings, the system attempts to recognize storytelling opportunities, situations in which some story is relevant.

This is an inversion of the standard way that retrieval problems are conceived. Instead of asking "What retrieval cue should the system build to represent a particular situation?" SPIEL

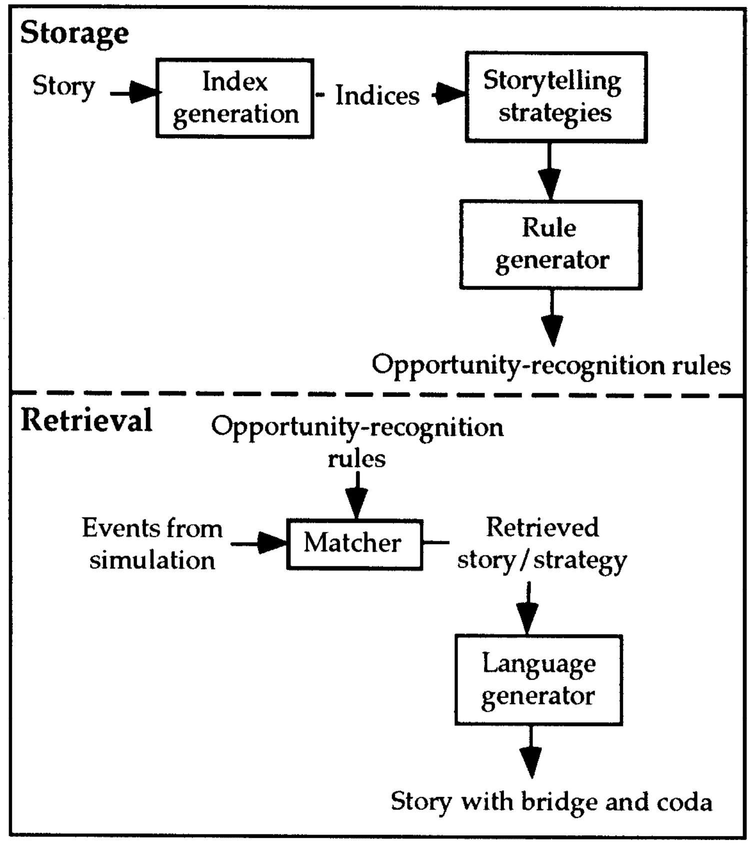

Figure 1.1. How SPIEL works

asks "What storytelling opportunity already characterized does this situation resemble?" This is analogous to the solution for natural language understanding found in the Direct Memory Access Parsing (DMAP) method (Martin and Riesbeck 1986; Martin 1991). DMAP changed the traditional natural language understanding question from "What knowledge structure should be built to represent this text?" to the question "What knowledge structure already in memory should be activated in response to this text?"

1.7.1. How SPIEL works

SPIEL characterizes and recognizes storytelling opportunities in two phases. The first phase, called the *storage phase*, prepares the system for storytelling. The storyteller examines its stories in the light of its storytelling strategies and determines what kinds of storytelling opportunities it needs to look for. In the *retrieval phase*, the system tries to recognize those opportunities as the student is performing the task. A schematic picture of these phases of SPIEL is shown in figure 1.1.

The storage phase begins with the manual construction of indices. An indexer watches the story on video and uses an indexing tool to construct one or more anomaly-based representations of what it is about. Indices are then processed using the storytelling strategies. SPIEL determines, for each combination of story and strategy, how and when it might tell the story using the strategy, thus characterizing the storytelling opportunities that the story affords. The rule generator converts descriptions of these opportunities into procedures for recognizing them, in the form of rules that are compatible with the GuSS simulation.

The retrieval phase of SPIEL is a teaching module in the GuSS architecture, implemented by a rule-based system. The rules created during the storage phase are matched against on-going events in the simulated world, as the student takes actions and the simulated characters respond. Successful recognition of an opportunity causes a story and its associated strategy to be retrieved. In order to introduce and explain the system's remindings to the student, SPIEL creates natural language texts, called the headline, bridge and coda that tell the student why the story was retrieved and what connection it has to the current situation. As shown in the example, these texts are presented to the student along with the story.

1.8. The contributions of this research

This work builds onto existing models of case retrieval a notion of case utility that is not strict similarity or association, but incorporates task considerations, such as the pedagogical import of a story. Similar considerations enter into other communicative tasks, such as argumentation.[6]

SPIEL's indexing representation for stories evolved from the Universal Indexing Frame proposal made in (Schank et al. 1990). The representation was pared to its essential elements: the notion of anomaly, based on expectation failure, and the notion of explanation underlying the failure. This representation is sufficient for a retriever to determine the applicability of a story for a wide range of pedagogical purposes.

Also significant is the way that SPIEL goes about its retrieval. Under the case-based teaching architecture, the stories of practitioners are viewed as the curriculum for students to learn. Therefore, it is natural for a storytelling tutor to be implemented as an architecture that starts with stories and tries to figure out how best to tell them. Stories stored in memory do not passively await activation by other processes. They actively determine what features will be looked for in the learning environment. SPIEL seeks opportunities to tell its stories using its strategic knowledge of what kinds of situations make stories relevant.

While SPIEL's storytelling strategies are specially tailored to the problem of teaching social tasks, the general principles they exemplify are more general. The two basic contributions of stories to learning-by-doing are their value as explanations and as pointers to possible expectation failures. Every case-based teaching system will need to implement these roles, although retrieval strategies will vary according to the type of expectations that are important in a given domain and the type of explanations that students are likely to need.

Consider SPIEL's strategies that show alternatives, like **Demonstrate risks**. Since students in almost any kind of environment are going to be interested in outcomes, the notion of showing cases with different outcomes than the student's is a technique that any case-based teaching system can probably use. Some of SPIEL's strategies are more specific, such as those that concentrate on the activities involved in planning: **Demonstrate alternative plan**, for example. If students were engaged in a hypothetico-deductive task, such as diagnosing a fault in an automobile, their expectations would have a different character. Chapter 4 shows an example of a strategy called "Demonstrate alternative hypothesis" that shows students cases that question their hypotheses. It would be applicable in a diagnosis domain. The explanatory strategies in SPIEL are also fairly specific to the exercise of social skill: they try to explain the thinking behind other people's actions. The principle is to make it easy for students to understand processes that are difficult to observe. Any learning environment that contains complex systems could use similar strategies to explain mechanisms that the student cannot directly see.

SPIEL's retrieval architecture has three basic parts: (1) the indexing of cases based on their inherent features, (2) the a priori application of presentation strategies to characterize case

---

[6]See (Ashley and Rissland 1987) and (Shafto, Bariess and Birnbaum 1992) for examples of case retrievers for argumentation.

presentation opportunities, and (3) the intervention of the tutor based on recognition of these opportunities. This framework applies to any case-based teaching system, whatever its cases and presentation strategies. In my Conclusion, I discuss progress towards an interactive case-based teaching tool that isolates the domain-independent aspects of this architecture for possible use in other programs.

1.9. Overview of the dissertation

This dissertation describes SPIEL in detail, concentrating especially on its storytelling strategies, indices and retrieval mechanism. The next chapter examines the GuSS architecture of which SPIEL is a part, looking at the social simulation and the role of stories within it. It includes an extended example of student interaction to show how stories fit into the educational goals of YELLO.

Having established the role of stories in the GuSS framework, I look at stories themselves in Chapter 3 and talk about how they can be labeled for the purposes of educational retrieval. This chapter expands on the notion of anomaly-based indexing introduced above, and also introduces the Story Indexing Tool that simplifies the manual creation of indices.

Storytelling strategies lie at the heart of the problem of building a storytelling tutor. The strategies and their implementation are introduced in Chapter 4; each of the four categories of strategies is thoroughly discussed in turn in Chapters 5 through 8. These chapters show how the storytelling strategies characterize storytelling opportunities, using different kinds of knowledge and inference mechanisms.

Chapter 9 delves more deeply into the implementation of the GuSS simulation and the problem of recognizing storytelling opportunities in a rule-based manner. The final step in the storage phase, the generation of recognition rules, is also examined here. Chapter 9 also shows how these rules operate in the retrieval phase: how they interact with the GuSS simulation and perform retrieval. Chapter 10 concludes the dissertation with a discussion of related work on case retrieval and case-based teaching, and future work on a case-based teaching tool derived from SPIEL.

2. SPIEL's teaching environment

2.1. GuSS and YELLO

SPIEL is the storytelling component of the learning-by-doing architecture known as **Gu**ided **S**ocial **S**imulation or GuSS. This chapter gives a brief description of the operation of GuSS to give a feeling for the system and to explain the role of storytelling in it; the system as a whole is described elsewhere (Kass et al. 1992; Kass et al. 1993; Blevis et al. in preparation).

As emphasized in (Kass et al. 1993), a central objective of GuSS is to engage the student in the realistic practice of social skills in context. The central unit of instruction is therefore the *scenario*, an extended interaction in which the student works on a realistic problem. In the course of a scenario, the student is confronted with a range of obstacles and opportunities typical of the real-world social task.

2.1.1. A YELLO scenario

The examples here and throughout this dissertation are drawn from the GuSS application called YELLO that teaches account executives at Ameritech Publishing the fine points of selling Yellow Pages advertising by letting them practice selling to a simulated client. The task is to get to know the client's business, come to some understanding of its market and its advertising needs, construct a proposal geared to these needs and to the client's concerns, present that proposal in a convincing way, and get the customer to buy. This section illustrates that task using a scenario in which the student is selling to a roofing contractor, Swain Roofing.

The cast of characters for Swain Roofing includes:

Ed Swain: the owner of the company,
Lucy Swain: the office manager, also Ed's wife, and
Dave Swain: the Swain's son, who is gradually taking over the business.

As is typical for small family-owned service contractors, the operation is run out of the Swain's home. Swain Roofing currently has a small ad in their local Yellow Pages directory. The underlying business situation, which the student may or may not uncover, is that the Swain's primary business, residential roofing, has been undercut by lower-cost, lower-quality competitors. The Swains are trying to get more business in the area of commercial roofing, where they feel their high-quality approach will be more valued. They are also interested in expanding their business into nearby areas they have not traditionally serviced. Ed and Lucy also want their son, Dave, to take over the business, and worry that it might not be strong enough for him to make a good living. Ed has given Dave responsibility for the firm's advertising, and Dave has started to look into cable TV ads as a way to get to potential customers.

A salesperson who asks good questions and discovers these concerns may be able to sell Swain Roofing a large ad campaign, including advertising in a "business-to-business" directory, a larger version of the current ad, small ads in headings other than "roofing," and display advertising in one or two directories in adjacent areas. A student who is less capable may have to settle for renewal of the existing ad.

2.1.2. A path through the Swain Roofing scenario

The Swain Roofing scenario begins with the student receiving the account information for the company, with an appointment already set up to visit the Swains. The student goes to the Swain's house for the appointment, and greets Lucy at the front door.

Student:	Hello. My name is Mike Johnson. I'm with Ameritech Pages Plus, the Bell Yellow Pages. I spoke with you on the phone about handling your account this year.
Lucy:	Please come in, Mike. Ed knows you're coming, and should be here shortly.

Figure 2.1 shows the appearance of the screen after this exchange. A picture of the scene, the Swain's kitchen, is at the right. Below this area is a picture of Lucy Swain. Above her is the text of what she has most recently said. Under her picture are emotion display meters that stand in for the multitude of cues to emotion given by people in social situations. Here the meters show that Lucy is being polite: someone has just come to her house, so she is looking somewhat happy (the first scale indicates happy/angry), somewhat interested (the second scale is interested/bored), and somewhat calm (the third scale is calm/nervous).

The action constructor, through which the student acts in the simulation, appears on the left. Here the student makes choices from menus and sub-menus to create utterances and other types of actions. Mike's greeting was actually the result of four action constructor choices, appended to each other:

1. Under the "courtesy" menu, there is an entry for "greet." Choosing this gets the English phrase, "Hello."
2. Under the "tell about" menu, selecting the "self" submenu and the option "name" gets the next part of the utterance, "My name is Mike Johnson."
3. The description of Mike's job, "I'm with Ameritech Pages Plus, the Bell Yellow Pages," comes from the option "affiliation" under the "tell about self" submenu.
4. Finally, to refer to the appointment set up in the previous conversation, the student can look under the "tell about self" submenu, and choose the option "reminder of appointment." The sentence is "I spoke with you on the phone about handling your account this year."

See figure 2.2 for a snapshot of the action constructor in use. Here the student is preparing to select "affiliation" from the "tell about self" submenu. The English language versions of the student's choices so far appear in the preview box at the bottom of the action constructor area. When the student finishes constructing an utterance, pushing the "Say It" button communicates it to the characters in the simulation.

Although Ed is the owner of the business, Lucy has an important role. An experienced salesperson would try to take advantage of Ed's absence to gather information about the business from her. However, a novice might not realize that Lucy could be an important source of information. Here the student engages her in small talk until Ed arrives.

Student:	What a wonderful view of the lake you have!
Lucy:	Thank you, we like it.
Student:	I'll bet your family enjoys fishing?
Lucy:	That was one of the major reasons Ed fell for the lot—he plans to be on the lake every chance he gets.
Student:	What do you fish for out here, trout?
Lucy:	No, they stock the lake with muskies. The hatchery's at the end of the road.
Student:	Muskies? They're a good game fish, right?
Lucy:	That's right.

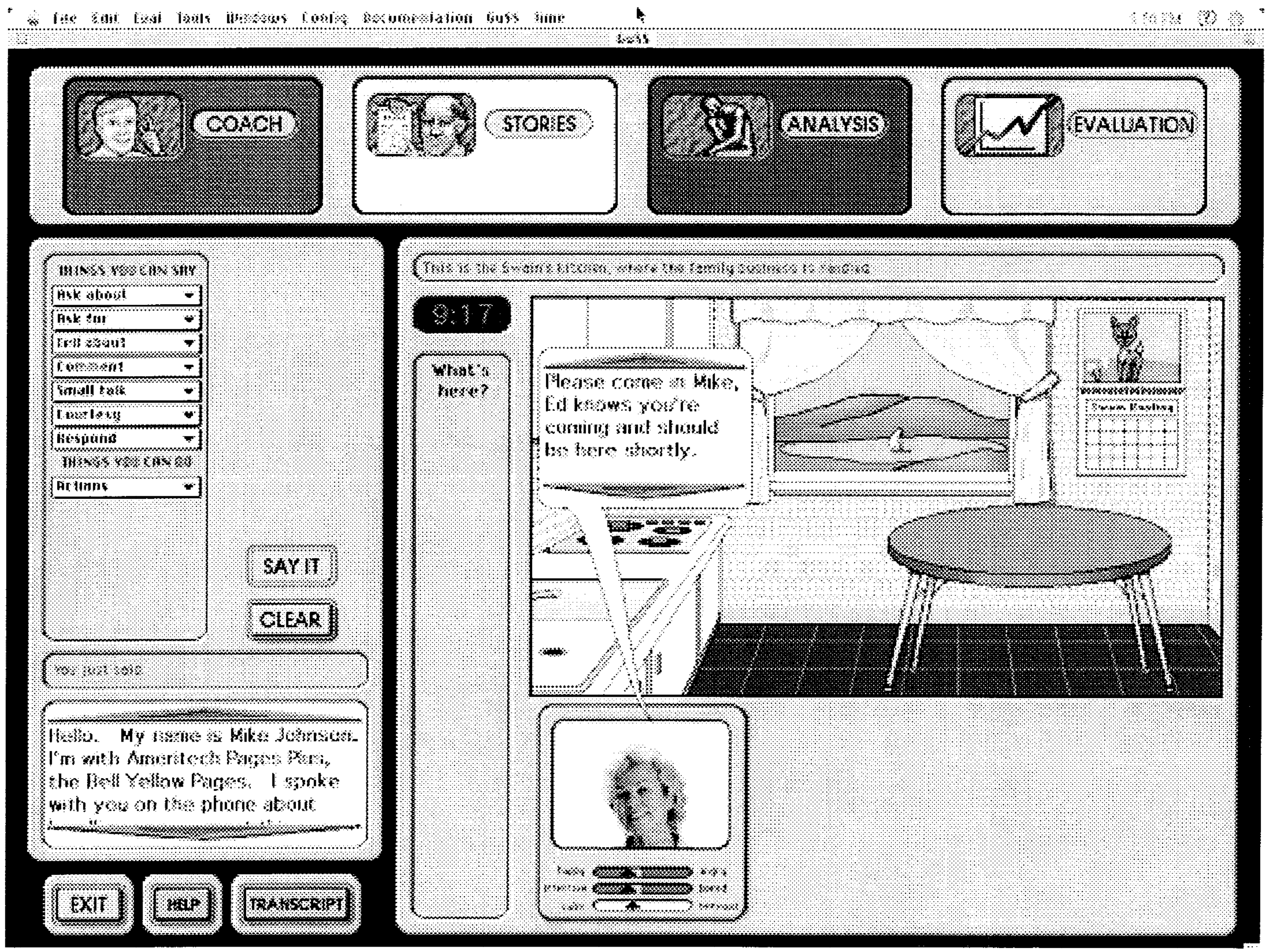

Figure 2.1. Lucy Swain greets the student

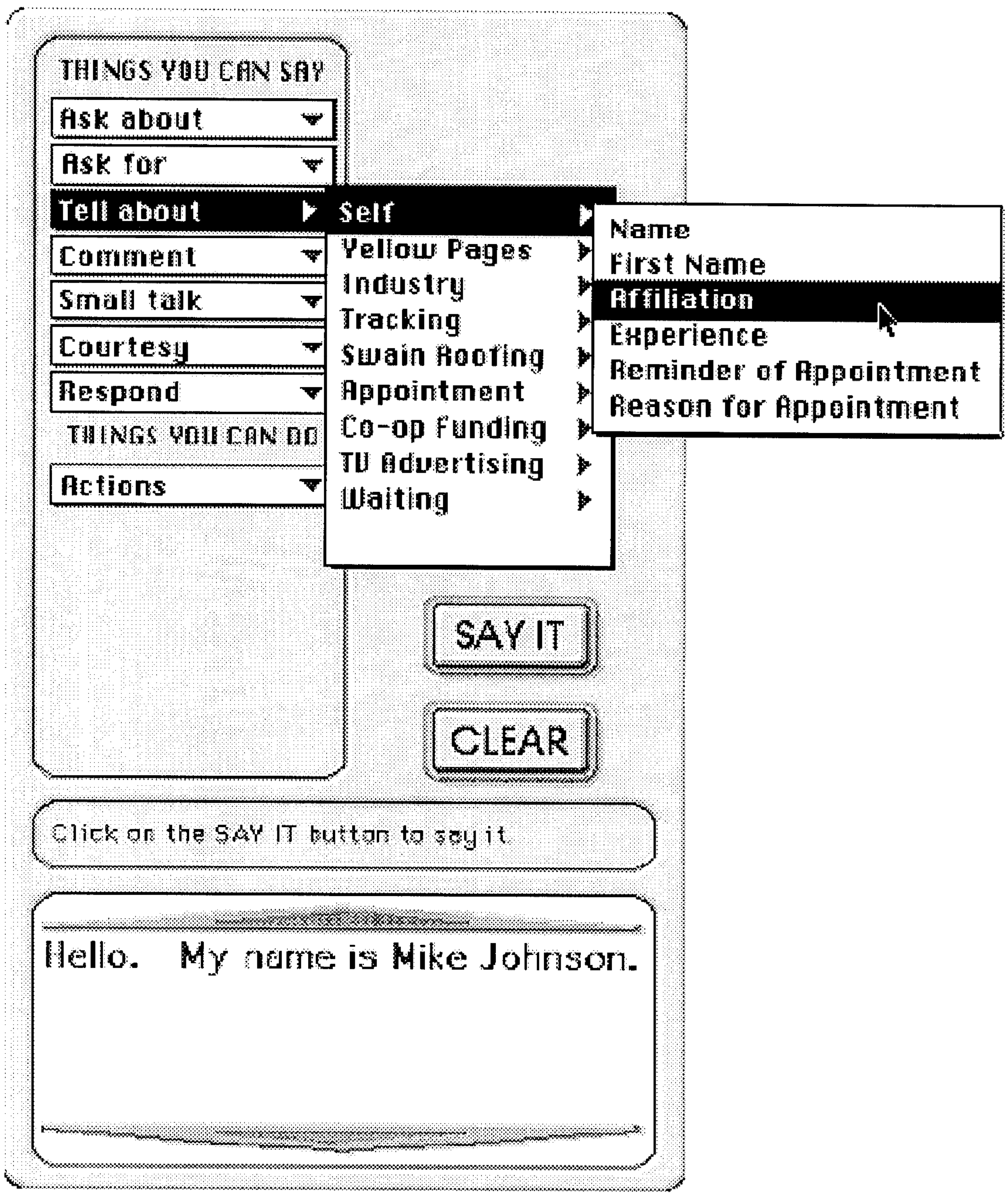

Figure 2.2. Using the action constructor

Headline:

A warning about something you just did

Bridge:

If you assume that Mrs. Swain will not have a role in the business, you may be surprised. Here is a story in which an account executive made a similar assumption that did not hold:

Story (from video):

"I went to this auto glass place one time where I had the biggest surprise. I walked in; it was a big, burly man; he talked about auto glass. So we were working on a display ad for him. It was kind of a rinky-dink shop and there was a TV playing and a lady there watching the TV. It was a soap opera in the afternoon. I talked to the man a lot but yet the woman seemed to be listening, she was asking a couple of questions. She talked about the soap opera a little bit and about the weather.
It turns out that after he and I worked on the ad, he gave it to her to approve. It turns out that after I brought it back to approve, she approved the actual dollar amount. He was there to tell me about the business, but his wife was there to hand over the check. So if I had ignored her or had not given her the time of day or the respect that she was deserved [sic], I wouldn't have made that sale. It's important when you walk in, to really listen to everyone and to really pay attention to whatever is going on that you see."

Coda:

An assumption that a spouse will not have a role in the business may be unrealistic.

Example 2.1. Telling "Wife Watching TV" story.[1]

2.1.3. A storytelling opportunity

At this point, Ed arrives. His picture appears in the scene, next to Lucy's.

Lucy: Well, there's Ed now. Hi honey, this is Mike Johnson from the Yellow Pages.

Since Lucy is deferential to Ed, the student has missed an opportunity to find out her thoughts about the business. From now on, he will be dealing with Ed. At this point, it is safe to conclude that the student does not expect Mrs. Swain to be useful in providing information about the business, or possibly, to have any role at all in the sale. This is a good time therefore to give the student some guidance indicating that the opportunity to hear Lucy's perspective has been missed.

SPIEL has a story that is relevant to this situation. When an opportunity to present a story appears, SPIEL signals the student by popping up and flashing its associated button, one of the four teaching module buttons across the top of the screen. The highlighted button contains a *headline*, a brief text indicating what kind of story is available. Here SPIEL highlights the "Stories" button with the comment "A warning about something you just did" as in example 2.1. See figure 2.3 for the screen at this point.

[1]Story from interview with Amber McLean, video clip #305.

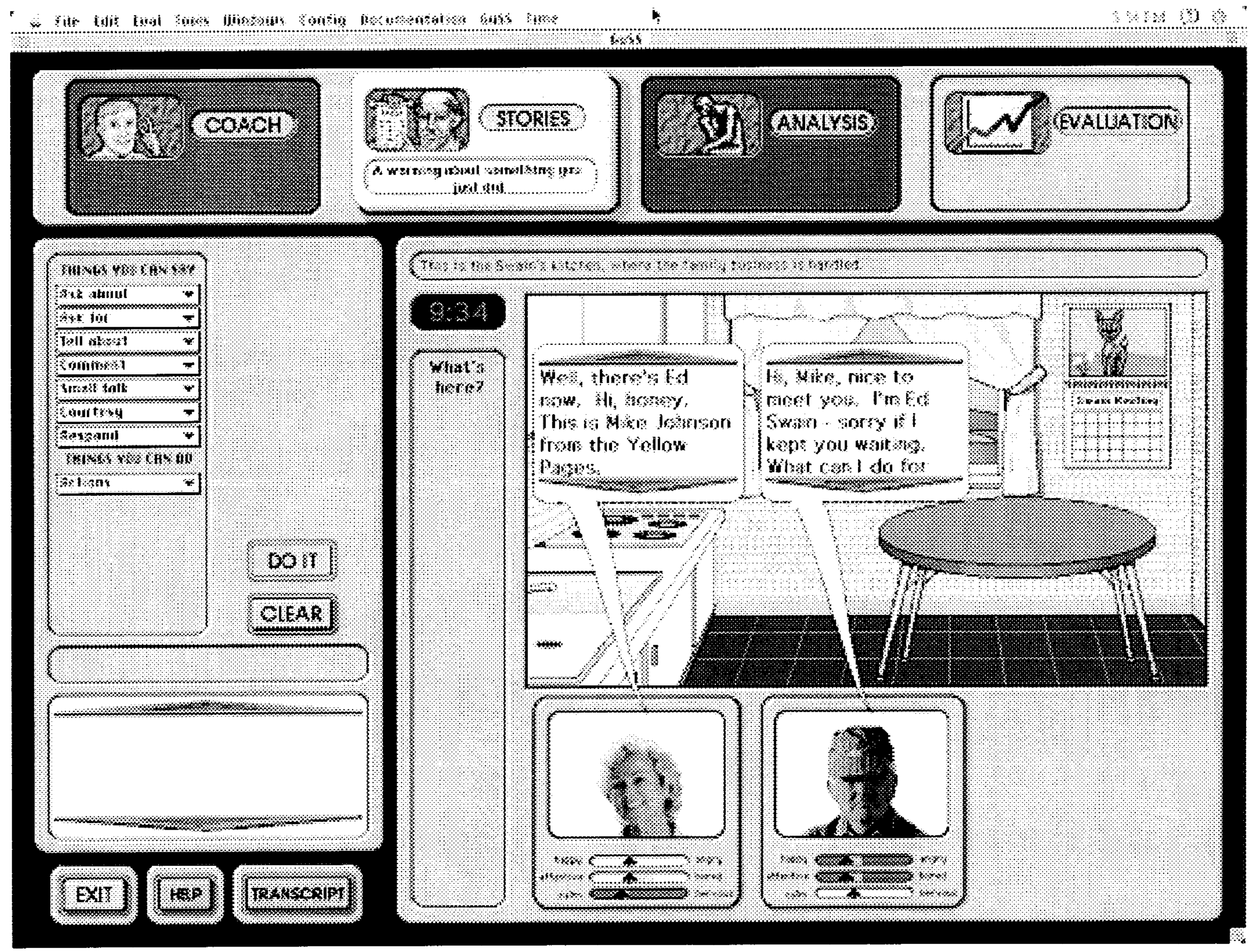

Figure 2.3. The Storyteller announces a story

If the student presses the button, the Storyteller screen is shown as in figure 2.4. The first item on the screen is the *bridge*, a short explanation of why the story has come up. The student can use the buttons below the video frame to view a video of an Ameritech account executive telling a story from her own experience. Example 2.1 shows a transcription of that story. Following the story, SPIEL sums it all up for the student with a short *coda* that describes a lesson of the story as it applies to the student's situation.

This example illustrates the synergistic interaction between simulation and storytelling. Without the story to provide the impetus to examine the situation, the student might never realize that an opportunity had been missed. However, without active engagement in the simulation, the student might lack the motivation and context to understand and remember the story.

This example shows the **Warn about assumption** storytelling strategy at work. In the story, the salesperson is surprised because she did not expect the wife of the "big, burly man" to participate in the sale and she did. The student's behavior indicates there might be a similar assumption about Lucy Swain at work. The **Warn about assumption** strategy calls on SPIEL to use a story about an incorrect assumption to show that an analogous assumption of the student's may be wrong.

After the student hears the story and closes the Storyteller window, the sales call can continue.

Ed: Hi Mike, nice to meet you, I'm Ed Swain—sorry if I kept you waiting. What can I do for you?

Student: Since I haven't handled your account before, Ed, I wanted to talk with you about what changes have taken place since last year....

2.1.4. Another storytelling opportunity

The student continues questioning Ed and finally wraps up the sales call:

Student: Is there anything I forgot to ask about the business?

Ed: You've covered it — why all the questions this year?

Student: It's important for me to know where you want your business to go in the upcoming year, Ed, so that we will have a better understanding of how to help you reach your goals. What are your long-term goals?

Ed: To make Dave work more so I can spend more time fishing!

Student: Thanks for your patience in answering all my questions.

Ed: OK. See you later.

The student has failed to seize an opportunity to probe into Ed's future plans for his business. Ed treats the question about long-term goals lightly, but his answer suggests many follow-up questions the salesperson should ask: What is Dave's future role? How close is Ed to retirement? This failure on the student's part is another opportunity for SPIEL. It brings up a story (shown in example 2.2) about a salesperson who makes the most of a similar situation.

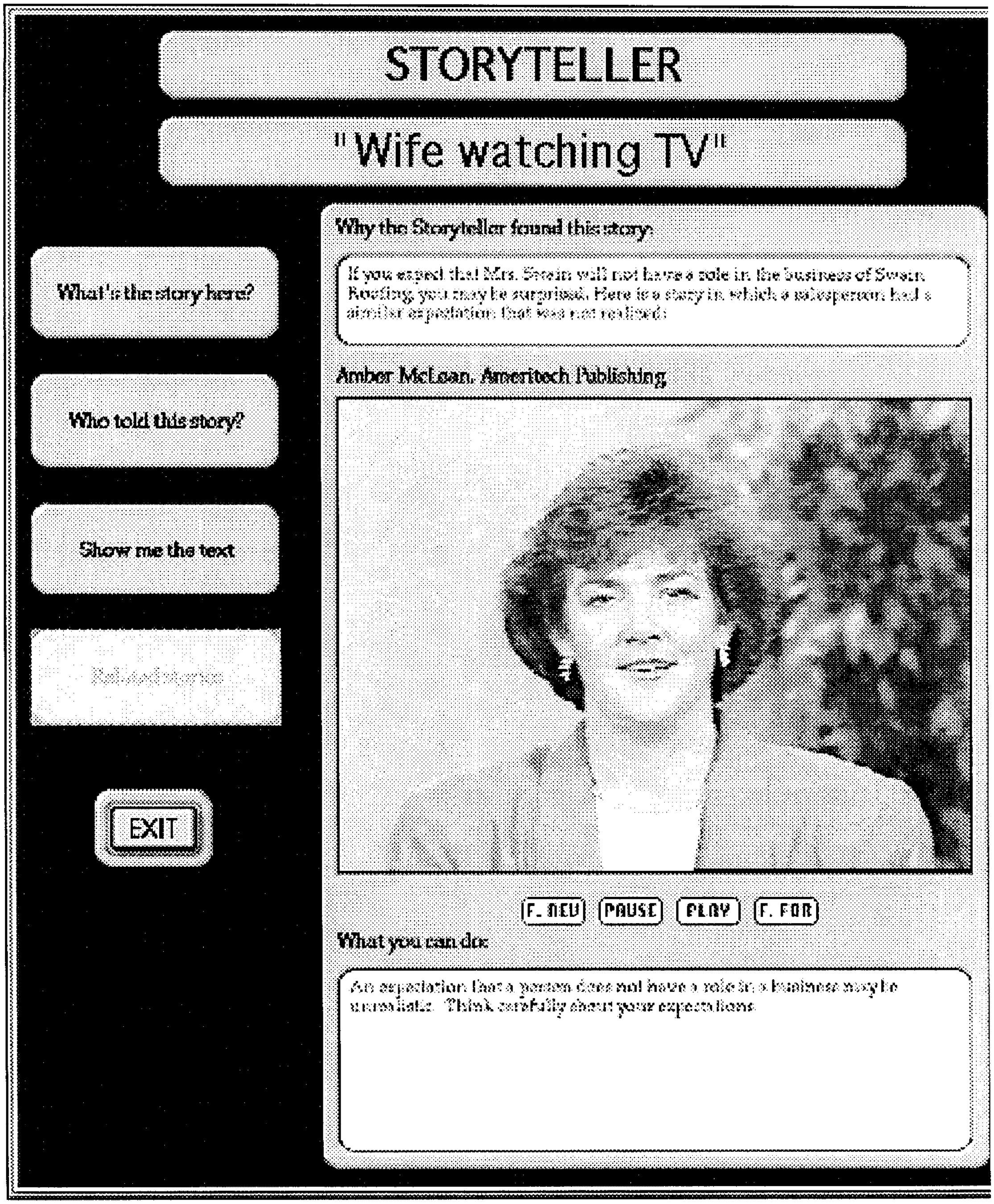

Figure 2.4. The Storyteller screen

A story about a different approach you might try

You gathered information about long-term goals in a cursory manner without much success. Here is a story about a similar situation in which an account executive used a different method and was successful.

Recently, I made a call to an electric contractor in Springfield, Ohio. One of the greatest things that a Yellow Pages person ever hears is "Leave it like it is," that person is a very good customer. But it's also a guy who needs a little incentive of they should buy, or maybe even a little reinforcement as to why they're buying. This company is a small company owned by a father, his son and just one other partner, basically a three-person operation in a small town. They have a very large ad; they have a quarter page ad with four colors. At that classification it's probably the largest ad in the directory. As a matter of fact, I did a history of all the ads from the past three years in that classification and their ad and almost every other ad in that classification had remained the same.

So when I got in there he had already told me that he wanted to leave it like it is, but he gave me the sense that he wasn't really sure why he kept buying it. So what I did was ask him flat out, I could sense that something was bothering him, I asked him. I said, "John, what's bothering you?" He said to me he wasn't sure that the cost justified itself. He wasn't sure that the ad was actually paying for itself. He opened up to me for the first time, after that. It set him at ease and we started talking about how much value a customer is to him, each call, and how many calls he needs to make that a profitable ad.

We also got into ideas of goals of his in the next five years, with his son coming up in the business and he getting close to retirement. We started thinking about how much more business he would have to do so he could obtain those goals. My advice in this situation is if you can get a customer to open up about the business, he's going to tell you a lot more about it from a goals standpoint in the next five years. This customer ended up buying the next larger size ad up to increase his business in electric contracting so he could achieve the goal of having the business strong enough for his son to take over in the next five years. He also, by talking to him, realized that the value of the product was so significant, to have in the book. Just from a dollars standpoint, it didn't take that many calls be an actual cost-effective program for him.

You gathered information about long-term goals in a cursory manner. It didn't work well. In the future, you might consider probing deeply into the customer's long-term goals.

Example 2.2. Telling the "Long term goals" story[2]

The "Long term goals" story illustrates an alternative approach to a similar situation. The salesperson in the story used the customer's long-term goals as leverage in selling a larger ad. This is distinctly different from what the student has done. This is an example of the **Demonstrate alternative plan** storytelling strategy. When a student has just handled a social situation unsuccessfully, SPIEL can tell a story about a similar situation that shows how another salesperson was successful.

2.2. Building a social simulation

Educational, technical and practical constraints all entered into the design of GuSS. All simulations strive for *accuracy*, the property such that student actions have the same results as in real life. The differences lie primarily in the degree of *fidelity*: the "feel" of using the simulation. A flight simulator is one extreme, where every effort is made to make using the simulator exactly like performing the task in real life. At the other extreme are "simulations," such as that found in "adventure"-style computer games, where the user is allowed to make a multiple-choice decisions at pre-determined choice points.

[2] Alan Beam, video clip #193.

The example above shows some of the simplifications required to build a social simulation. It might be assumed that maximum fidelity is the ultimate goal for every simulation, but there are good reasons to aim for less than total duplication of real life. In many domains, not every action performed in the course of doing a task is worth teaching. A simulation that leaves out unimportant aspects of an interaction or includes them only in schematic form will save the student from spending time doing things that have no educational significance. A simulation should aim for fidelity in those areas where the student's skills are most important and where the student is likely to need to learn. For example, a Yellow Pages account executive spends a lot of time driving to customers' businesses and waiting in waiting rooms. We do not simulate these aspects of a salesperson's life in YELLO because doing so would teach little and reduce the amount of time available for exposing the student to interesting selling situations.

There are also technical aspects to the problem of fidelity, particularly in a social simulation. For the most part, simulations of physical phenomena can be made as close to "real" as needed, since equations describing the phenomena are usually known.[3] This is not true for social simulations. Many of the phenomena of interest in a social situation are not yet well enough understood for detailed modeling to be practical. To give just one example, GuSS does not admit free natural language input from a student although it would be much more natural than the action constructor. Research has not yet produced theories that account for human natural language understanding sufficiently well to enable a simulated character to understand such input.

While a technical apparatus, even a complex one like an aircraft cockpit, can be reproduced with great fidelity, the same level of fidelity is not currently possible for social situations. Attempts at virtual reality notwithstanding, computers cannot easily provide the full range of signals and cues found in the social world, nor can they recognize and respond in a realistic manner to the student's tone of voice, body language, etc. Radical simplifications of the environment must be made for a social simulation to be tractable. This is one reason why it is important to augment a social simulation with depictions of real world situations such as stories, which can compensate for this loss of fidelity.

The major simplifications made in building GuSS fall into four categories:

Physical action: A student's physical activity is present in GuSS only in schematic form: the student does not have to drive to the Swain's house, poke through a file cabinet to get account information, or pick which chair to sit on in the Swain's kitchen. These kinds of actions, while part of the salesperson's life, are not what students need to learn about.

Synchrony: The simulation does not have real time operation; it has to give students time to use the action constructor. Waiting for the student to act slows the simulation. In part, this also serves a pedagogical purpose; giving students time to think carefully before constructing their utterances in the simulated world should help them carry the habit of reflection to their real-time conversations. A negative consequence is that students do not feel pressure to think on their feet in the simulation as they would when confronted with a real person in conversation.

Natural language: Even though conversation is the activity that the student does most in a GuSS simulation, the intent of GuSS was not to solve the natural language understanding and generation problem. As shown in the example, the GuSS interface uses a system of hierarchical menus for students to construct natural language utterances by composing natural language phrases. Although the menus encode a finite number of options, the sheer number of

---

[3]Note, however, that for very complex physical systems there may be practical reasons to prefer qualitative rather than quantitative simulations (Hollan, Hutchins & Weitzman 1984).

possibilities forces students to use the menu system to construct something to say rather than to choose a completed utterance. The set of options also is dynamic: the more the student finds out, the more issues can be raised. Preliminary experience with GuSS suggests that after a short period of acclimatization during which students browse through the options in the action constructor, they begin to use it easily as a means to express themselves.

Visual realism: Because of the commitment in GuSS to an open-ended simulation, the program cannot carry on video conversations, complete with gestures, facial expressions, and body language. Every possible state must be recorded ahead of time in such a system.[4] Our compromise on this issue is the unrealistic, but useful, expression meters associated with each agent. Here the student can get a reading of the emotional expression of an agent. It is not the same as learning to read body language, but it does teach the student what features are worth learning to identify.

2.2.1. Open-ended simulation

Since GuSS is founded on learning-by-doing, one aspect of the social world that is of crucial importance is open-endedness. For some domains, a simulation with a small number of choices is acceptable, when in real life a practitioner has a fixed number of possible actions at any given time and the task is to pick the correct one. The range of possible actions in most social situations is very large and it may be difficult even to generate a set of reasonable actions. Picking the right answer out of a small set of predefined choices is simply not an activity that has a counterpart in the social tasks where GuSS has been applied. This is the rationale behind the design of the action constructor and of the design of GuSS's simulated characters.

The characters or *social agents* in the GuSS simulation operate in an autonomous, unscripted manner. They are, in effect, simple social planners. The agents observe events in the simulation and react to them based on internally-represented goals and plans. In particular, when they converse with the user and with each other, they are enacting plans to say certain things, plans that are realized in particular utterances.

Each agent is a self-contained production system, similar to those found in expert systems such as OPS5 (Brownston, et al. 1985). At each point in the simulation, the agent looks for the productions, or *decision rules*, that match its current working memory, which is called in GuSS the agent's *mental state* because it represents what the agent knows, believes, expects, etc. The rule that matches best is fired, which causes the agent to perform some action. For example, Lucy has a rule that causes her to respond to compliments about her house by saying: "Thank you, we like it." This rule matches when she observes Mike's comment, "What a wonderful view of the lake you have." Other rules cause changes to her mental state: she stops expecting Mike to arrive when she greets him, for example.[5]

With this design, agents are naturally extensible. If a new topic of discussion is needed, an agent can be given new decision rules to respond to that topic. Many of the decision rules that an agent needs are fairly generic: ones that any agent might have, or ones that can be strongly associated with a particular type of personality. Ed Swain expects that the salesperson will have some knowledge of terminology particular to his industry. He will get a little irritated if the student shows a lack of understanding of basic terms. This expectation would probably be common to many strong-minded business owners. In GuSS, those aspects of Mr. Swain's decision making that could be considered common to other such individuals are segregated into

[4]See (Stevens 1989) for an interesting example of a social simulation that operates in this way.

[5]GuSS's rule-based simulation of agent's decision-making is described in detail in Chapter 9, as a part of the discussion of SPIEL's rule generation mechanism.

a personality stereotype: *entrepreneurial spirit.* Ed Swain, by virtue of having this personality, inherits mental states and decision rules associated with the stereotype.

2.3. Combining instruction and practice

The GuSS simulation is a complex mechanism for generating realistic social interaction that supports students' learning-by-doing. However, teaching modules such as SPIEL remain essential. Interaction with the simulator gives students experience with only one set of possibilities: those that have been put into a scenario. This is not enough for students to gain a broad understanding of the task they are learning. The social world has tremendous variation and no amount of simulation would be enough for students to see all possible interactions or outcomes. This, of course, is true of learning-by-doing in all complex skills — the student can't experience everything.

In order to become truly expert in a field, a student must learn the abstract principles at work and how those principles apply in practice. Simulated practice is useful, but students who have only faced the problems presented by a simulated environment may simply memorize the actions that work in the program and be unable to adapt when faced with novel situations outside it. Formal schooling attempts to teach principles directly, but often fails to teach how they are applied. As a result, motivation is often low, and much of what is learned is quickly forgotten, or remains "unintegrated or inert" (Collins, Brown and Newman 1989).

GuSS combines instruction and practice, achieving a powerful synergism: students can reflect on their experiences in the learning environment with the help of the instruction given by GuSS's teaching modules, of which SPIEL is one, and the principles described by the teaching modules are motivated, operationalized, and made memorable by putting them to use (Schank 1991; Williams 1991).

The job of a teaching module is to be an advisor: to look over the student's shoulder, watch the simulation, and give relevant advice. The guidance that students receive in GuSS is provided by a library of teaching modules that observe the action in the simulation and intervene to present relevant material. In addition to SPIEL, the current implementation of GuSS has two other teaching modules: the Analyzer and the Coach, each presenting a different kind of material.

2.3.1. The Analyzer

While SPIEL holds the experiences reported by practitioners as a source of informal knowledge of a field, the Analyzer holds "textbook" knowledge: for YELLO, there are about 40 multi-media "textbook sections" relating to social practices, including information about personality types, communication tactics and organizational behavior. The Analyzer in YELLO also has information about standard problems in selling such as keeping a customer's attention, and dealing with particular classes of objections.

If the student encounters difficulty in keeping a customer focused on the sale, the Analyzer will intervene with a discussion of how to get and keep attention in selling. Like SPIEL, the Analyzer gives a brief introduction that explains to the student why it has intervened. The "keeping the customer's attention" section includes two videos of re-enactments of selling situations: one showing a salesperson who does very poorly at attracting the customer's attention, and a similar situation in which the salesperson is successful. The successful salesperson lowers his voice almost to a whisper when the customer turns away so that he must turn back to hear what is said. This technique is not something that the student can actually practice within the simulation, but it is an important part of a salesperson's repertoire. The re-enactments presented by the Analyzer can give students vivid examples of useful techniques even when they cannot be practiced in the simulation. There is a large amount of supporting information available in the Analyzer so that students who are interested can explore its issues in depth.

The Analyzer has the job of explicitly teaching principles and techniques, much as a standard textbook or lecture aims to do. Its multi-media capacity is obviously an improvement on these techniques, but the principal value of the Analyzer arises because of its association with GuSS. The Analyzer can present its material at the moment that the student is likely to be interested: embedding its discussion of principles in contexts in which they are directly relevant.

2.3.2. The Coach

The Coach teaches procedural principles of a social task. It has an abstract model of the goals a student should have and tracks those goals to see when they have been satisfied. In YELLO, the Coach uses a hierarchical decomposition of the steps in the process of selling Yellow Pages advertising and follows the student's accomplishments, letting the student see what important issues are outstanding.

For example, YELLO's Coach associates three important sub-goals with the phase of initial contact with the customer: information gathering, rapport building, and "seeding," giving the customer new ideas about Yellow Pages advertising. When the student hears Mr. Swain talk about his problems with the competition, the Coach signals that an important piece of information has been gathered, helping the student understand if the conversation is going in right direction.

A student who is unsure about any part of the task can query the Coach by selecting elements of the task decomposition and using buttons, such as the "Why?" button. If the student asks "Why?" about a goal, the Coach explains how that goal fits into the overall task of selling. If the student asks "Why?" about the achievement or partial achievement of a goal, such as a piece of information the customer provides, the Coach will explain the relationship of the event to the goal. So if the student asks "Why?" after selecting Mr. Swain's complaints about his competition, the Coach explains that information about Mr. Swain's competitive situation can be used to put together an ad program that clearly distinguishes his business from his competitors and that his concern about this issue can be used as leverage in a sales presentation. Students also have access to a "How?" button that calls on the Coach to give hints about general strategies for achieving goals, and a "Now what?" button that asks the Coach to give suggestions for what tactics to try next.

The Coach acts as a guide through what may be an unfamiliar process to many students. It lays out the general course of the selling process and makes explicit the relationships between its different steps. In negotiating this terrain, students can call on the Coach to remind them of why they are doing what they are doing, and to get hints about how to go about it. The Coach also records events that contribute to the success of various goals, so that students can check to see how they are doing. In this way, the Coach acts as a "smart notebook" that automatically records important information. The intent is that students will learn what things to pay attention to and will model their own note taking after that of the Coach.

Of all the teaching modules in GuSS, the Coach is probably the most similar to other types of tutoring systems, most of which have knowledge that is primarily procedural. It is perhaps closest in spirit to Elsom-Cook's "guided discovery tutoring" concept (Elsom-Cook 1990) and the coached practice environment of SHERLOCK (Lesgold, et al. 1992). What these systems have in common is their reliance on hinting and user control of the interaction with the tutor.

2.3.3. SPIEL

Stories are a good complement to the other forms of instruction in GuSS. The Analyzer explains and demonstrates some of the general principles that are important for the task, but these descriptions need to be supported by real-world corollaries if students are to find them believable and understand their application. Re-enactments are an important part of the Analyzer because they show how salespeople put general principles into action, but students are aware that these vignettes are staged for their benefit. Students trust the "reports from the field" that stories provide. First-person stories also put students inside the action, giving them a glimpse of a salesperson's way of thinking, something that cannot be easily achieved in a re-enactment.

Students can get hints from the Coach about how to accomplish certain goals, but the Coach cannot be more than mildly directive, because it only knows the general outlines of the task. To make the Coach truly directive is beyond the scope of GuSS. It would require a complete theory of social behavior and persuasion: a research project in itself. Stories can be examples that illuminate and make concrete the general hints and suggestions given by the Coach.

2.3.4. The GuSS Shell

Teaching modules are combined with the simulation in the GuSS Shell, which is the basis for all GuSS applications. The Shell contains three major components: the **Interface Manager**, which handles input from the student and displays information to the student; the **Simulation Engine**, the general-purpose social simulation mechanism that manages the simulated world; and the **Teaching Modules**, which monitor the state of the simulation looking for opportunities to present useful guidance.

Figure 2.5 is a schematic diagram of the Shell, showing the interconnections among these three components and their associated data. The student interacts with the Interface Manager, which passes the student's actions to the Simulation Engine. The engine computes the state of the simulated world, making use of the production system model of each agent, and using the location and artifact definitions that are specific to the particular scenario the student is in. The Simulation Engine sends messages back to the Interface Manager telling it to display changes in the simulated world. While this interaction is occurring, the Teaching Modules examine the state of the simulation, signaling the Interface Manager when they have something to present to the student.

2.4. Teaching modules vs. intelligent tutoring systems

GuSS may appear superficially similar to an intelligent tutoring system (ITS), the standard means of building computer-based tutors. ITS programs contain a tutoring component that, like a teaching module, monitors students' task performance and intervenes to provide guidance (Polson and Richardson 1988; Wenger 1987; Sleeman and Brown 1982). The significant differences between GuSS's tutoring and that of ITS systems have to do with the kind of knowledge that the tutor has and the way that the tutor intervenes.

An intelligent tutoring system is composed of three modules: the expert module, the student model, and the tutor. The student model observes the student's behavior in the system and diagnoses the state of the student's knowledge. The expert module solves the same problem as the student using expert knowledge. The tutor compares the student's state of knowledge as represented in the student model against the expert standard and determines where the differences lie. These differences are what the student needs to learn. The tutor then decides how to intervene to correct the student.

ITSs have been constructed to teach geometry (Anderson, Boyle and Yost 1985), introductory computer programming (Reiser, Anderson and Farrell 1985), and arithmetic (Burton 1982). They have been notably less successful in teaching complex skills[6], such as medical diagnosis (Clancey 1982). Lesgold and Lajoie (1991) note that even the technical skills involved in

---

[6]Complex in the sense of (Funke 1991).

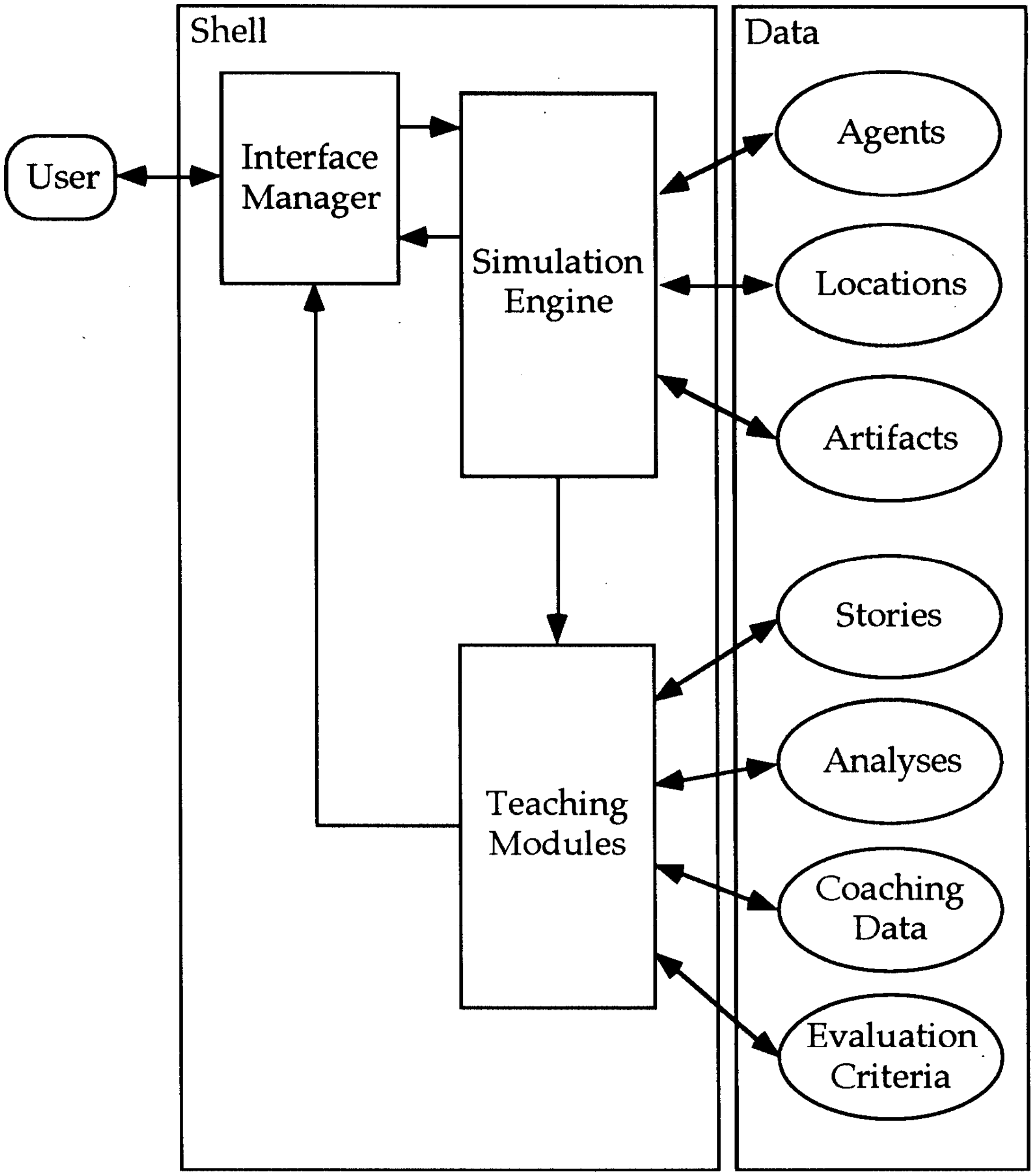

Figure 2.5. The GuSS architecture

electronics troubleshooting turned out to be too complex for the standard ITS framework to be useful.

There are four sources of complexity in GuSS's domain, as shown in table 2.1. These complexities have important consequences for applying the ITS model to the problem of tutoring for social skills. The large quantity of existing student knowledge that students bring to the learning of social skills makes the construction of a student model problematic for these tasks. It might be argued that students learning geometry do have a lot of experience with lines and figures. However, there is an important distinction to be made here. A student learning geometry is trying to fit his or her intuitions about geometry into a new form of discourse involving the geometric proof. In social arenas, the discourse is not new: beginning account executives using YELLO will have encountered salespeople throughout their lives, and

Table 2.1. Differences between social skills and those typical to ITSs

	ITS domains	Social skills
Existing student knowledge	small	very large
Complexity of task	low	high
Solution procedure known	yes	no
Agreement among experts	yes	no

probably have experience in sales already. YELLO cannot teach as though its students were encountering the conceptual basis of selling for the first time. For students in YELLO, learning how to fit new concepts and new skills into their existing repertoire is at least as important as learning the new material.

There is also the problem of exactly how expertise can and should be characterized. In domains of social expertise, the complexity of the task and the lack of a known solution procedure mean that there are often as many right answers as there are experts. This fact calls into question the whole pedagogical thrust behind the ITS approach. The next sections examine each of these issues.

2.4.1. The student model

Student models come in many different forms. vanLehn's (1988) survey lists three different dimensions on which models differ: the bandwidth of information that can be obtained from the learning environment, the type of knowledge representation used in the model, and the kind of student-expert differences that are allowed. What these models have in common is that they are complete and dynamic, representing the whole of the student's knowledge as it relates to the task, and tracking the state of the student's knowledge as that knowledge grows and is refined by interaction with the tutor.

Student modeling systems have been constructed for elementary tasks, such as addition and beginning computer programming, where there is a small amount to learn and therefore a small number of possible misconceptions. It is significant that a student's knowledge can differ from the expert standard in only a small number of ways, because for most of the models that vanLehn discusses, the system must have a complete catalog of all possible "bugs" or incorrect bits of knowledge that the student might have. For this reason, most intelligent tutors are designed to handle students who are complete novices. However, a social skill, such as negotiation, is refined from the social expertise students already have. Students learning a social task never begin as novices. They bring to bear a great deal of prior knowledge and experience.

Because everyone's social experience is different, the number of possible misconceptions a student can have is very large. No bug library could encompass all possible misconceptions a student might have accumulated over a lifetime of social experience. It is also unlikely that a tutoring system could automatically build a detailed model of a student's idiosyncratic social knowledge through observation, especially given that this kind of knowledge acquisition remains an unsolved open problem in expert systems (Boose 1986).

The problem of modeling social knowledge does not hamper SPIEL's tutoring, because what SPIEL needs to model is not students' state of knowledge but their state of interest. What SPIEL uses instead of a dynamic student model is a much more static and schematic set of expectations about the student's preconceptions. Static student models have been considered insufficient for ITSs, but they will work for case-based teaching because what students are interested in does not change as fast as their state of knowledge (Edelson 1993).

2.4.2. The expert module

In addition to a model of the student, the ITS architecture also requires a model of expert performance (Anderson 1988). This model has two roles in ITS systems: it is a source of the tutorial material that is presented to the student, such as the correct way to proceed in a particular problem, and it is the standard against which the student is compared. For this second role, it is essential that the expert model contain all of the knowledge necessary to perform the task correctly. Anderson notes various ways that this can be accomplished. One method uses a "black box" model that merely produces the correct result without its internal states bearing any relation to how a person might perform the task: for example, a computer's calculation routines can be used to check a student's math. Another possibility is the rule-based expert system where the rules are intended to correspond to heuristics that the student should learn. Finally, there is the cognitive model, where the processes that the system goes through are intended to mimic the mental processes of an expert reasoner. Tutors with these models try to train the student to go through the same cognitive processes as the expert.

One might well argue about the assumptions underlying these characterizations of expertise and the educational methodology they imply. Recent work from a constructivist perspective takes issue with the notion that knowledge can be transmitted in the way that the ITS model implies. (See (Lave and Wenger 1991), (Clancey 1991) and (Newman, Griffin and Cole 1989) for views of education and expertise that challenge the traditional formulation on which the ITS model of education is based.) I will raise just one point that is relevant for SPIEL's tutoring. In social domains, experts of equivalent ability may disagree about the right way to solve a problem because they have different personalities and different personal styles. Even in areas of technical skills, Lesgold and Lajoie (1991) found that different experts possess different "dialects of expertise" that cause them to approach problems differently. The expert systems community, which has as its aim the construction of models of expert problem-solving, has found that reconciling the views of multiple experts in a single rule-based system is very difficult (Boose 1988). Teaching modules in GuSS cannot therefore have a traditional rule-based expert model as the standard against which students are compared, without losing their ability to represent the whole range of expert knowledge and experience.

The expert model that SPIEL does have is in some sense a cognitive-type model in the sense that Anderson describes. The stories in SPIEL's story base form a case-based model of expert social problem-solving. The way that the stories are organized for recall reflects the important issues of expectation failure and explanation that students must master, and they reflect the experts' understanding of the social world. What is different in SPIEL is how this model is used.

2.4.3. Style of teaching

Where SPIEL differs most from intelligent tutoring systems is the kind of material that it presents and the kind of impact it aims to achieve. In the ITS paradigm, the comparison between the student's behavior and what is predicted by the expert model is intended to indicate defects in the student's understanding. The tutor intervenes when the student has diverged from what the expert model predicts, and its role is to get the student back on track. The intervention provided by ITSs is either procedurally-based, intended to get the student to understand and follow the correct procedure for solving a problem, or fact-based, intended to give students familiarity with a body of facts. Such teaching is less applicable in a social arena, where there often are no hard-and-fast rules and few useful universal facts.

The case-based view of expertise implemented in SPIEL calls for the tutor to play a quite different role. Rather than tutoring knowledge that describes the correct solution procedure or some collection of facts, SPIEL has a set of stories, a range of experiences that bear on the problem at hand. Instead of saying that the student is in error, the tutor can say “Here’s a

Table 2.2. Issues in case-based teaching and their realization in DUSTIN, Creanimate and SPIEL.

	DUSTIN	Creanimate	SPIEL (YELLO)
Domain	Business English	Biological form and function	Selling techniques
Retrieval	Direct association, single strategy	Cue formulation, multiple strategy	Opportunity recognition, multiple strategy
Indices	Links to scenes	Retrieval conditions	Story content
Communication	None needed	Same representation	Translation mechanism

situation in which what you're doing turned out to be a bad idea." Instead of saying that the student should perform a certain action, it can point out a situation in which that action led to a good result.

The system can leave it to the student to judge whether the advice is relevant, and if so, how to apply it. The system does not have to guarantee that the student gets it right. If a student in the Swain Roofing scenario does not manage to make a sale to Ed Swain, the learning experience in YELLO is not greatly diminished. GuSS will be successful even if students do poorly in the simulation because they will come away having been exposed to some important cases and seen how they apply in particular contexts. Such students have begun to build the case base of experience on which their expertise will be founded.

2.5. Case-based teaching

Building students' case bases is the goal of every case-based teaching system. However, there are many different ways that this idea can be implemented. For example, in addition to Creanimate (Edelson 1993), the tutor for elementary biology introduced in chapter 1, there is also DUSTIN (Ohmaye 1992), a tutor for English as a second language. DUSTIN uses cases that are re-enactments of situations in which English is used in a particular context. For example, in one scene in DUSTIN, the student faces the problem of answering the necessary questions to get through Customs at Chicago's O'Hare Airport. A student who has trouble performing this task can watch a re-enactment in which someone else talks to the customs officer, and thereby see a case of correct behavior that can be emulated.

DUSTIN, Creanimate, and SPIEL are three contrasting implementations of the case-based teaching idea. The differences between them illustrate some of the important issues in building case-based teaching systems. Table 2.2 shows some of these contrasts with respect to three issues: retrieval architecture, indices for cases, and communication between environment and tutor.

2.5.1. Retrieval

Retrieval is simple in DUSTIN. The student has a limited set of scenes to go through; along with each scene is associated a re-enactment that the student can choose to view. No retrieval processing or reasoning is required of the system itself. The creator of the system links each case to its associated scene, doing all the work of determining what case should arise at which point. DUSTIN's retrieval scheme is successful because its cases serve only a single purpose, showing students the correct way to use English in a particular context.

The hard-wiring approach works for a single scenario and a handful of cases, but it does not lend itself to the storytelling tasks found in Creanimate and SPIEL. Unlike re-enactments, stories can be relevant in many different circumstances. If each story is labeled according to the point in the scenario where it should be retrieved, it will be twice as difficult to build two scenarios that use those stories as it would be to build one: labeling would not carry over.

Creanimate improves on hard-wiring by performing retrieval using a cue formulation approach. At certain stages in the conversation, such as the point when the student has suggested a modification to an animal, the system puts together a query and performs memory search to retrieve the case that best matches the cue. Creanimate uses cases in three basic ways: to exemplify biological issues that students raise, to show interesting similarities among animals, and to show cases that violate standard expectations about animals. Each strategy formulates a cue to search for stories that meet its particular requirements. For example, if the student decides to build a fish that can dance, the exemplar strategy looks for examples of other animals that dance.

Cue formulation retrieval means that Creanimate's case libraries and its dialogue plans can be developed independently. If Creanimate is given a case of a new kind of animal, all that is required is that the case be indexed, and it will be accessible from any dialogue. New dialogue plans can retrieve relevant cases simply by formulating cues and making retrieval requests at appropriate points.

SPIEL's retrieval architecture is an alternative to the cue formulation approach. Its strategies are used to characterize storytelling opportunities in advance, so they can be sought during the course of the student's interaction. What practical difference does this make? In comparison to Creanimate, it seems to make little, because Creanimate's stories are already indexed by their retrieval conditions: much like the recognition condition descriptions that SPIEL computes.

2.5.2. Indices

The difference between Creanimate and SPIEL lies in the systems' indices. Creanimate uses a tutoring-specific indexing vocabulary. Cases are indexed according to the pedagogical purposes that they can serve. A video of a mantra ray using its wing-like fins to swim is indexed by the fact that it is an exception to the rule that fish do not have wings. This is because the system has the expectation violation strategy that calls on it to retrieve cases that include such violations. This video might also be a good example to use to show off the use of fins in swimming. Before the case can be accessed by the exemplar strategy, another index must be added to represent the anatomical form and biological function demonstrated in the video.

By contrast, SPIEL's indices are derived without reference to the system's pedagogical techniques. They are a function of the stories themselves. An indexer can describe a story for the system's benefit without thinking about what educational role the story might have in a learning environment. The content-based indexing scheme in SPIEL allows easier experimentation with storytelling strategies. A corpus of stories does not have be re-indexed if a new strategy is added to the system's repertoire.

2.5.3. Communication

Because the case-based teaching architecture is composed of two parts, the learning environment and the tutor/retriever, these modules must be able to communicate. In his work on the Teaching Executive, Jona (in preparation) identifies three ways this can happen:

1. The tutor and the learning environment can share the same representation of the task. The tutor can therefore recognize whatever concepts it needs to identify through reference to the state of the learning environment.
2. The tutor can have a representation for reasoning about the student task that is separate from the learning environment, translating between representations in order to perform recognition.

3. Characteristics of the learning environment that are important for the tutor's purposes can be manually labeled by their tutorially-important properties, and those properties can be communicated to the tutor when they arise.

Option 1 has been implemented in Creanimate. Its case retriever used indices that refer directly to the knowledge structures that the system uses for its dialogues. There is therefore no translation problem in moving between the representation of the task and the indices of cases. If the student is carrying on a dialogue in the learning environment about an animal that uses claws to catch food, for example, the retriever can get directly to cases that are indexed by the concepts "claws" and "hunting." The difficulty with option 1 is that there is no possibility of modularity between tutor and learning environment. The only way to construct another storytelling tutor, even one using the same cases, is to build another integrated system.

SPIEL uses option 2. This architecture has the advantage, as I have argued, that it allows SPIEL to characterize storytelling opportunities in terms that are independent of a particular learning environment. A tutor can therefore be a separate module. This modularity does have its drawbacks: translation between tutor and learning environment is a knowledge-intensive process. The tutor must have a way to identify, in the learning environment, instances of any concept that it is interested in.

Jona argues instead for the third alternative, where the learning environment is labeled manually with terms that will be meaningful to the tutor. This solution resembles the ITS paradigm, except that instead of relying on an explicit model of student and expert knowledge to specify points of intervention at run-time, it uses a manual labeling of the task environment using the user's own implicit understanding of right and wrong.

The problems with this approach are similar to those associated with ITSs. When labeling the environment, there is no provision for reconciling different dialects of expertise (Lesgold and Lajoie 1991) and differences in personal style that lead experts to disagree about whether a particular action should be labeled right or wrong. There is no obvious way to bring to bear the system's knowledge of its own resources, the way that SPIEL takes advantage of its story base as a case-based model of the domain.

These issues are not critical in the well-specified procedural tasks to which the Teaching Executive has been applied. It is not clear that this approach can be extended to the complex domains where stories are most important. One aspect of Jona's approach that is important to consider, however, is the use of knowledge acquisition from the user to replace difficult reasoning on the part of the tutor. This same idea can be applied to the problem of knowledge acquisition in SPIEL: see Appendix C.

2.6. Storytelling in GuSS

Storytelling is a type of tutoring that is well-suited for advising students learning complex social tasks in a simulated environment. SPIEL presents stories told by people who are experts in the areas the student is studying, stories that are drawn from the expert's own experience. Such stories are particularly valuable in a simulated environment, since students can see how the simulated situations they come across reflect the world of real practice.

The way SPIEL teaches is also important. It draws the student's attention to the associations it makes with others' experience, but it does not try to tell the student what to do. Since SPIEL is not required to present one "right" answer, it can show multiple perspectives on difficult issues. It might, for example, bring up two stories, one about someone who was successful doing what the student is doing and another about a failure in the same situation. It is important for students to recognize that even experts can disagree about the best course of action.

I have argued that the traditional approach of intelligent tutoring systems would be intractable if applied to the task of building teaching modules in GuSS. However, the task of a case-based teacher can be performed without having a complete model of the student's problem-solving process and without having a model which is the standard of expert performance. What a case-based teaching system such as SPIEL needs is a model that predicts where the student's interests will lie. This model is encapsulated in SPIEL's storytelling strategies and its indices.

One of the critical issues for any case-based teaching system is the design of an indexing representation. What a retriever can do with cases will be largely determined by how it can access them. SPIEL's indexing representation captures intrinsic properties of stories, from which it derives possible tutorial uses for them. This kind of representation stands in contrast to other case-based teaching systems whose indices directly incorporate an assessment of a case's pedagogical utility: such as the direct links between scenes and cases found in DUSTIN, or connections between cases and expectations they violate in Creanimate. SPIEL's modular approach allows experimentation with a range of pedagogical strategies because different strategies can be tried out on the same body of indices. It also places less of a burden on the indexing step, which in case-based teaching systems is performed manually. SPIEL's indexing representation is the subject of the next chapter.

3. Labeling Stories

3.1. Introduction

For a storyteller to choose a story to tell, it must know what its stories are about. It has to know that a story will convey a certain message to the student: that it will present an interesting alternative, for example. Its knowledge of stories must be highly organized and efficiently used. In picking a story to tell, the storyteller should be like a librarian that can point out books worth reading without having memorized the text of every book in the library. If a patron says, "I'd like a nature book on bears for my ten-year-old," the librarian can determine what kind of book would satisfy the need, and can use a catalog that describes books in appropriate terms to locate a few good candidates.

There are two kinds of knowledge such a librarian needs: knowledge about what kinds of books are useful for what purposes and labels associated with the books themselves to say what they are about. A storyteller needs similar knowledge. It has to understand the tutorial functions its stories can serve, and it has to know what stories can serve those functions. The first kind of knowledge is embodied in SPIEL's storytelling strategies. They are discussed at length in Chapters 4 through 8. This chapter describes SPIEL's labels or *indices*[1], descriptions that summarize stories for the purposes of tutorial retrieval, representing what stories are about well enough so a storyteller can tell decide which one to tell.

3.2. Labeling for case bases and story bases

The problem of designing an indexing representation is not new or unique to storytelling. Any system that must retrieve chunks of knowledge can benefit from a representation that summarizes each chunk. Even the contents of books are frequently indexed so that a reader can jump to the portion of the text that is likely to be of interest.

The indexing of knowledge has become increasingly important in artificial intelligence as researchers have moved from systems with knowledge in the form of small, easily-described chunks such as rules, to systems that reason with larger, more complex chunks of knowledge, such as the cases used by case-based reasoning (CBR) programs.

The need for indexing is prominent in CBR systems because their knowledge sources, cases, are complex entities packaged as a single unit. Researchers in case-based reasoning realized early that a system cannot search every item in memory while deciding what to retrieve (Kolodner 1980). It would be intolerably inefficient to examine of the contents of each case in order to select good candidates. Cases have to be labeled and these labels have to be organized so that search for an appropriate case will be efficient.

3.2.1. The indexing problem

The need for an organized vocabulary of labels for cases gives rise to what is known as "the indexing problem," the problem of designing an indexing representation (Schank 1982; Domeshek 1992). An index has to represent what is important about a story for the purpose of retrieval, but not all purposes have the same requirements. A retriever interested in a historical account might want to retrieve stories that describe events that happened on a certain date. A

---

[1] Index is the term that has been traditionally used in case-based reasoning for the label given to an item stored in memory by which it can be retrieved. This has produced a certain amount of confusion with some of the other uses of the word, such as the "inverted index file" in information retrieval. I follow Owens' example (Owens 1991) and use the term "label" interchangeably with it.

linguist might want to retrieve stories in which the speaker says "you know..." more than five times in one minute. These tasks call for quite different representations.

The most important demand that an index must satisfy is **utility**: it must encode features that are predictively useful for the task. The features that appear in an index should help the retriever predict if a case will be a good solution or if a story will make a useful point. Often such features have the property that they are important in explanations of how a solution matches a problem (Schank 1986). For example, the case-based cooking planner CHEF (Hammond 1986) uses "taste" as a feature to label recipes because taste is part of the goal being satisfied when cooking. If CHEF is called upon to produce a certain dish with a spicy taste, spicy recipes are probably good candidates, even if they use different ingredients, because the spices that a recipe adds to a dish largely determine its taste. On the other hand, whether a recipe requires the use of a sauté pan or a casserole is less important for CHEF since the use of certain utensils is not part of its goal in cooking and doesn't usually contribute to the taste of a particular dish. "Taste" is a useful index for CHEF; "utensils used" is not.

The second criterion is that an index must contain features that are **computable**: available at the time of retrieval. This criterion is often difficult to reconcile with the previous one. For example, the most succinct and useful feature for a medical case is probably the diagnosis of the patient's disease, but such a feature would not be useful for a case-based medical diagnosis system. The system would be unable to retrieve candidate cases for diagnosis without creating the diagnosis first. Symptoms, on the other hand, are more easily gathered, but not as concise or predictive.

The demands of utility and computability frequently oppose each other, since the most useful and predictive features are often those that are expensive to extract or ones that require solving the problem first. Creating an indexing representation for a particular CBR system requires making productive tradeoffs between these two constraints. CHEF, for example, can use cooking goals as indices because they are highly predictive and they are part of its input, requiring little computation. IVY (Hunter 1989), a case-based medical diagnosis system, must start from symptoms, which are less predictive, and must employ additional reasoning to get to cases. It uses observable symptoms to retrieve a set of abstractions, and applies additional discriminating tests to infer indices that it then uses to retrieve candidate diagnoses.

3.2.2. Indexing in a story base

SPIEL borrows from the case-based reasoning tradition, but the difference in task between case retrieval for problem-solving and story retrieval for tutoring is significant, and has significant consequences for SPIEL's index representation. The employment of stories in SPIEL is different from how cases are usually used in case-based reasoning. The standard problem-solving CBR system retrieves cases in order to re-use old solutions for new problems. It has one question to ask of a case: can it solve my current problem? A storyteller is more like (Kolodner and Wills 1993) case-based system for creative design; it has many possible uses for a given story, many storytelling strategies.

SPIEL also differs significantly from CBR systems in that the contents of its case base are stories in video form, not representations. Video stories are *opaque* to SPIEL: it cannot reason with them. Some case-based reasoners dip into the representation of a case when more detailed knowledge of it is sought. IVY, for example, looks at features of its cases when indices are not enough to discriminate between possible diagnoses. SPIEL cannot do this: the retrieval of stories must be based entirely on the contents of the index.

These differences require that SPIEL's indices be richer than those typically found in CBR systems. They have to contain all of the information needed for retrieval since there is no case representation to fall back on. SPIEL's information needs are greater, because the system has multiple uses for its stories and needs a range of different details to satisfy the criteria of each different kind of utility.

3.2.3. Explanation and indices

The index must contain everything that a retriever needs to know about a story. For SPIEL, this means that it must say enough about a story that its fitness for different storytelling strategies can be determined. One easy way to satisfy this criterion is to encode the events of a story literally in the story's index (Osgood and Bareiss 1993), to have an event-level representation of the story's contents. Everything the story says can be encoded this way, therefore nothing will be left out. This is an appealing approach since it means that writing down the representation is straightforward: the index describes what the story says. Let's apply this approach to the following example of a reminding:

> "Once while watching the demolition of a building in Chicago, I was struck by how ineffectively the work was being done. The wrecking ball hit one of the concrete supports near its center again and again with little result. It was frustrating to watch the lack of progress.
>
> This poorly-executed demolition reminded me of the time I saw a bull-fight in Spain. The matador kept dealing out blows to the bull with his sword with seemingly little effect. The failure to execute a "clean kill" made the whole affair grotesque." (Schank et al. 1990, 7-8)

A event-level representation (adapting liberally from [Osgood and Bareiss 1993]) of the "incompetent matador" story would look something like this:

> The main event of the story was a matador stabbing the bull and being unsuccessful in killing it.

While this summarizes the story well, it does not go far in explaining the reminding. Consider a similar summary for a different story:

> The main event of the story was a two-year-old trying to tear a plastic bag and being unsuccessful at tearing it.

Both stories describe failed destruction actions. The reason that the matador story is a better analogy to the wrecking ball story has to do with the point of interest of the story. What makes the story interesting is not the actions of the matador but the expectations of the person observing. The observer expects that professionals will be effective at their jobs, but both the matador and the crane operator fail. The story about the two-year-old is not as analogous because there is no such expectation for two-year-olds.[2]

To understand why this should be so, we must look more closely at the phenomenon of reminding. As discussed in (Schank 1986), explanation is one of the key processes for improving one's understanding of the world. Remindings help by bringing to mind past explanations that may contribute to understanding a new situation. In this interpretation, the matador story comes to mind because the teller is trying to understand why the crane operator is failing to do the job. People place a great deal of trust in professionals such as doctors, civil engineers, or even equipment operators. The teller wants to know under what conditions he can count on professionals to do their jobs correctly. The matador story, because it is an

[2]Note: It is always possible to argue that a reminding is useful. There will always be some way to "tweak" (Kass, Leake and Owens 1986) a recalled case into something useful. A rough notion of the relative goodness of cases can be estimated from how much tweaking is required to make each useful. Under this metric, the story about the two-year-old is less useful than the matador story.

ineffective attempt at destruction carried out by a supposed professional, provides another example to compare with failure of the crane operator.

In an educational context, the matador story would probably not be a great story to tell because it does not explain anything about the matador's action. Suppose, however, that the story was more explanatory:

> This poorly-executed demolition reminded me of the time I saw a bull-fight in Spain. The matador kept dealing out blows to the bull with his sword with seemingly little effect. *It turned out that he had been wounded in a previous bullfight and was afraid to get close enough to really finish the job.* The failure to execute a "clean kill" made the whole affair grotesque.

This version of story does supply a possible explanation for the crane operator's failure. It raises certain explanation questions (Schank 1986) such as "Is the operator being unduly cautious, due to some previous accident?" or perhaps "Does the city have overly-restrictive rules about crane operation due to fear of lawsuits?" A learner, in attempting to answer such questions, can come to a better understanding of the original situation.

An instructional storyteller is seeking this kind of explanation transfer. Students are presented with stories so that they will better understand and explain the problems they face (Schank 1990a; Schank 1990b). To be effective, the storyteller must know not just what is interesting about a story (the gruesome scene at the bullfight), but why it is interesting (the failed expectation about professionals). It is only through reference to the "why" that the storyteller will be able to determine what counts as a similar "what." The story about the two-year-old has a similar failed destruction scene, but does not have any relation to the expectation and does not help explain its failure as readily.

Event-level representations tend to leave out the "why" part of a story, because it is also omitted by storytellers, who rely on their hearer's background knowledge to fill it in. The expectations "I thought a professional would be good at his job" and "I thought the bull would die quickly" do not appear anywhere in the events of the story, but readers easily infer them. People are successful in understanding stories even when they are extremely oblique. Consider the following story (from the film *Manhattan*, cited in [Schank 1990b, 40]):

> "She's seventeen. I'm forty-two, and she's seventeen."

Schank argues that this is a story. The absence of normal story conventions, such as plot, has little effect on our ability to interpret it. In fact, the paucity of data is a cue that tells hearers to fill in what is missing with their own background knowledge. We assume the standard things about seventeen-year-old girls and 42-year-old men and interpret the story accordingly.

This is obviously an extreme case, one where a direct translation of the story into a event-level representation would fail to capture anything about what makes the story interesting. In fact, no events are stated. It is the standards against which the story is judged that make the story interesting at all. In a culture where women marry early and age disparity is not considered important, there might be nothing anomalous or memorable about the *Manhattan* story, but in twentieth-century America, it is a violation of cultural standards and therefore arouses interest.

Even though this example is extreme, the point it makes often holds. People who tell stories do not mention those key features that make a story interesting because they can rely on the background knowledge that their hearers bring. Therefore, for educational purposes, an index must be more than a summary of a story. It must be an interpretation of a story whose creation, by necessity, entails the use of thematic knowledge, just as story understanding does (Schank and Abelson 1977; Wilensky 1983). The index must explain what is interesting in a story for a particular purpose, and why.

3.3. The Universal Indexing Frame

SPIEL's indexing framework had its starting point in the Universal Indexing Frame (UIF), introduced in (Schank et al. 1990). The UIF was an attempt to create an indexing framework for stories such as those in SPIEL's story bases. It was developed by observing story remindings in context and developing a framework within which remindings could be explained. Work on the UIF began with the insights I have described in this chapter:

- the importance of intentionality: the need to represent causality due to the actions of an intentional agent;
- the centrality of explanation: the role of explanatory coherence in reminding; and
- the anomaly: the representation of what is interesting in a story as a function of the violation of expectations.

The UIF was meant to be universal, in the sense that the widest possible range of stories could be indexed using its organization and vocabulary. Like any such universal scheme, it succeeded in some ways, but not in others. Where it succeeded best was in representing stories about social behavior that people find to be interesting. This is not surprising since the material we worked from in the development of the UIF was precisely this type of story. The UIF is given a thorough description in (Schank et al. 1990). I recount here those aspects of the representation that are a good starting point for understanding SPIEL's indexing system.

The UIF has three main parts: the content grid, the context grid and the global slots. Figure 3.1 shows the content grid for an index for the building demolition story from (Schank et al. 1990). The grid shows four perspectives on the events in the story. Each perspective is distinguished by three values: the *viewer*, the person who holds the perspective; the *view*, the type of outlook held by that person; and the *agent*, the person whose actions are viewed. The first column therefore describes how "self," the person watching the demolition and narrating the story, has perceived his own reaction to the situation. The second column is the expectation that the narrator had of the crane operator. The third column is how the narrator perceived the operator as acting. The last column, the anomaly column, records the salient differences between the columns: in this case, the unexpected result.

3.3.1. Content grid

The content grid holds a set of explanations for events in the story. Each explanation is a frame with 17 slots, so the explanations are organized into an N x 17 matrix. Each column contains an intentional chain (Schank and Abelson 1977), an explanatory chain of reasoning and behavior. It starts with an intention that leads to action that leads to some result. In the UIF, this chain had 14 pieces that can be roughly divided into four parts: pre-action, action, post-action, and analysis.

The pre-action slots indicate the state of the agent before the main event described in the story. In the example index, the narrator is curious about the on-going demolition before he begins to observe it. This is called *anticipatory affect* in the UIF. There is a similar slot for beliefs, *pre-task belief.*

The action slots explain the genesis of an important action in the story. The action is construed as fulfilling some cognitive *task*, such as plan execution in the example. It arises due to a *theme* that the agent holds that gives rise to a *goal.* The agent tries to achieve the goal by using a *plan.*

The post-action slots describe the outcome of the plan. The *result* is the direct outcome of the action, usually some impact on the agent's goals. In the example, the plan perceived as being used by the operator (column 3) has the result of achieving the destruction goal, but slowly.

Slot names	*View1*	*View2*	*View3*	*Anomaly*
Viewer	Self	Self	Self	
View	perceived	expected	perceived	
Agent	Self	Operator	Operator	
Anticipatory affect	curious			
Pre-task belief				
Task		execute	execute	
Theme				
Goal		demolition	demolition	
Plan				
Result		fast demolish	slow demolish	failed expectation
Positive side-effect				
Negative side-effect				
Resultant affect	bored			
Post-task belief				
Delta affect	-invert			
Delta belief				
Trade-off				

Figure 3.1. UIF index for the building demolition story from (Schank et al. 1990).

Positive or *negative side-effects* show the other impacts of the plan. Finally, *resultant affect* and *post-task belief* can be compared to what the agent felt and believed previous to the action.

The analysis section lists three kinds of comparisons. The *delta-affect* and *delta-belief* slots compare the belief and affect slots, before and after, to arrive at a description of a change that occurred in the story, such as the negative inversion of affect from interested to bored in the first column. The *trade-off* slot compares the result and side-effects slots to see the overall impact of the action.

In each column of the content grid, these fourteen elements serve to explain a perspective on the events of a story. For example, the second column in figure 3.1 gives the narrator's expectations. The narrator expected the crane operator to achieve the quick destruction of the building when taking action to achieve the demolition goal,. This chain of reasoning uses only three slots in the frame, because it is a fairly simple expectation.

3.3.2. Context grid

The second component of the UIF is the context grid. The context grid displays the relationships between all of the individuals described in the story. Since people often have different perspectives on a situation due to their relationships with each, this information helps make sense of the different perspectives given in the content grid.

Figure 3.2 shows the context grid for the demolition story. The relationships are read as "column person" has "relationship" with "row person." So, the entry in the second row, third column is read as "the building is being destroyed by the crane operator."

3.3.3. Global slots

Finally, the third main component of the UIF is the two global slots, the setting and the anomaly. The setting describes the environment in which the story occurs. Setting was regarded as a surface feature of a UIF, and since the UIF was an attempt to build a general domain-independent framework, the details of what should and should not be included in the setting were never developed.

The anomaly slot singles out the most salient element in the anomaly column. In the index in figure 3.1, there is only one contrasting row, the difference between expected result and actual

Thematic Roles	Self	Crane operator	Building
Self		observed-by	observed-by
Crane operator	observer-of		destroyed-by
Building	observer-of	destroyer-of	

Figure 3.2. Context grid for the demolition story from (Schank et al. 1990).

result. It becomes the main anomaly by default. Stories with more complex UIFs can have more than one point of contrast but they will only have one main anomaly, one main point. Of course, a story can be interpreted in different ways and have more than one main point. The story would then have multiple UIF indices, one for each point.

3.3.4. Indexing in ABBY

A parallel but not entirely independent development in the representation of indices for stories was the indexing scheme in ABBY (Domeshek 1992). ABBY is a case retrieval system designed to give advice to the lovelorn in the style of "Dear Abby." In ABBY, intentional chains and the relationships between them are central as in the UIF, but implemented more flexibly. Domeshek allowed a variety of relationships between chains beyond the notion of anomaly in the UIF. ABBY allows the attachment of causal and aberrational links between any parts of any intentional chains. For example, one person's effect (ABBY's version of result) may interact with someone else's plan.

3.4. Experience with the Universal Indexing Frame

Work on SPIEL originally began using a corpus of stories about selling consulting services. Approximately 160 of these stories were indexed using the UIF. Three lessons from this experience informed the development of SPIEL's indexing system:

- Complex anomalies are rare. The anecdotes that salespeople told generally showed only one salient contrast between an expectation and reality.
- Explanations could be simpler. For stories about action in the business world, the belief, emotion, and analysis parts of the UIF were rarely important.
- The context grid was insufficient. Two-place relations between actors were not enough to capture all of the interesting relationships in stories.

These lessons were applied to the development of a considerably-simplified version of the UIF that was used for YELLO. The most significant simplification was to reduce the complexity of the content grid. I found that, to capture the notion of expectation failure, it was enough to have one column that showed a perspective contrasted against one column that showed what actually happened. SPIEL's indices could be limited to two columns, one of which would always be an actual column with the same agent as in the perspective column. This simplified content structure reduced the number of between-column differences to be recorded in the anomaly column, usually to just one. The anomaly column was therefore eliminated because it had no more information than could be found in the global anomaly slot.

The columns themselves were also simplified to better fit the stories. From the original 11 slots in the UIF, SPIEL's indices use six: theme, goal, plan, result, positive side-effect and negative side-effect. These six are the core of the intentional chain in planning and acting (Schank and Abelson 1977). Domeshek also found that these were the most important elements of the intentional chain from the UIF.

Domeshek, however, embroidered parts of the UIF, building more elaborate explanation structures, to capture more precisely the causality of ABBY's stories. For example, he added an additional slot between goal and plan, called "goal->plan," that indicates how a plan for a goal was arrived at. This slot holds information such as "this plan might be the default one to use for the goal," or "this is the only plan that met a certain resource constraint." ABBY could therefore find similarities regarding how a plan was chosen as well as what the plan was. SPIEL does not incorporate these additions, not because they do not contribute to a full explanation of the anomaly, but because of the second constraint on indexing representation: availability at retrieval.

SPIEL is trying to draw analogies between stories and student situations. It must be able to compare what it knows about a story with what it can determine about the student. It is hard enough to detect that a student is using a particular plan, and much harder still to make a reliable determination as to why the student chose it. To compare the reasoning behind a choice made by a character in a story with the thinking behind a student's choice, SPIEL would have to understand of the reasoning a student goes through in choosing a plan. Since students' social knowledge is likely to be very large in extent and quite idiosyncratic, a complete understanding of it cannot be built simply by observing a brief interaction with the GuSS system. The difficulty in using a feature such as "goal->plan" at retrieval time argues against its inclusion in an index. The theme, goal, plan, result and side-effect aspects of the intentional chain are things that can be fairly readily observed or inferred without a substantial commitment to a detailed model of each student's reasoning.

I found it was also possible to simplify the structure of results and side-effects. These three slots all capture impacts on goals. The original UIF treated them all as more or less the same, the only difference being how central they were deemed to be. Whatever the creator of the index considered the most important effect would be the main result, the others would be side-effects. In SPIEL's indexing frame, I have removed some of the ambiguity surrounding results and side-effects. Since a plan in an index is enacted to achieve the goal found in the same column, I consider its success in this regard the primary result: the result slot is always an impact on the main goal.

Side-effects also had a certain ambiguity in the UIF. They could be used to show the costs associated with a main result, for example, that a plan was achieved only at tremendous cost. They also had a second function to show how the result achieved by one person impacted others. A side-effect of a customer saving money by cutting back on expenses might be that a salesperson fails to make a sale. This second role of side-effects turned out to be the most important one for SPIEL. So, in SPIEL's indexing frame, side-effects are always interpreted as effects on the goals of the *viewer* rather than the agent. For example, a column in one of SPIEL's indices shows a salesperson (viewer) who hoped that a customer (agent) would like an ad's design (result, impact on customer's goal), inclining him towards buying it (positive side-effect for the salesperson's goal of getting a sale).

To allow relationships to be represented in more ways than the context grid allowed, it was replaced with a social setting slot. This slot contains a list of the social relationships relevant in the story, letting the system represent a notion like partnership, instead of as a two-place relation between business partners, as a larger relation, between all of the partners and the business that they jointly form. It also allows two people to have more than one relationship at the same time, a circumstance that could not be represented in the context grid. Domeshek also found that the context grid was too restrictive and allowed a wider variety of social relationships to be represented in ABBY's indices.

3.5. SPIEL's indexing frame

The anomaly is the central organizing principle in SPIEL's indices. Having simplified the UIF to fit SPIEL's stories more closely, I realized that the anomaly concept should be considered primary, the place where the creation of an index begins, rather than an emergent property of the contrasting intentional chains. The other parts of the index support and explain the anomaly. The general form of an anomaly represented in SPIEL can be stated as follows:

"Actor X had an expectation that Y would happen, but actually Z happened."

This interpretation meshes well with the original notion of an anomaly (Schank 1982) as an expectation failure that demands explanation. A failure requires a contrast between an expectation and reality. In the "building demolition" story above, the narrator wonders explicitly why the demolition was so ineffective. He expected a "clean kill" but it didn't happen. SPIEL's representation would encode the anomaly as follows:

> Observer assumed that the operator would knock the building down, but actually the operator failed to knock the building down.

The previous chapter showed an instance where the student was presented with the "Wife watching TV" story. It has the following as its central anomaly:

> The salesperson assumed the wife would have the role of housewife, but actually she had the role of business partner.

This corresponds to the point of the story that emerges when the story is told in Chapter 2. The student is reminded that his assumption about Mrs. Swain's role may be in error in the same way as in the story.

3.5.1. Anomaly types

Assumptions are one kind of expectation that may fail, and hence one possible type of anomaly. SPIEL has five kinds of anomaly, based on five different kinds of perspectives that someone might have:

Assumed: An assumption is an expectation that will generally always be present in the given task and is probably subconscious. (Called **Expected** in UIF terminology.)

Example: The salesperson assumed the wife would have the role of housewife, but actually she had the role of business partner.

Feared: A negatively-valenced expectation, or fear, will be present when someone acts to ward off a threat to a goal. Fears turn out to be misplaced when the threat is not realized.

Example: The salesperson fears that the client will object to the price of a proposed ad, but actually the client doesn't.

Hoped: When a character has unrealized hopes, it is usually a case of over-optimism or a situation in which a plan is not well-executed. (Called **Wanted** in UIF terminology.)

Example: The salesperson hoped the customer would track how many calls the Yellow Pages ad was generating, but the customer wouldn't do it.

Standard: Every field has its own set of standard beliefs and procedures. Novices learn these through instruction, and then go out into the real world where they do not always apply. A standard-violation anomaly, instead of reading like "I expected X..." reads more like "You might expect that X...". (Called **Ideal** in the UIF.)

Example: You might expect that a salesperson would defend the merits of a proposal, but actually this salesperson agreed with the client's counterproposal.

Perceived: Stories sometimes highlight failures that result from misperception, such as reading something into a situation that is not there.

Example: A salesperson perceived the client to be interested in a larger ad, but actually the client wanted to spend less money on advertising.

3.5.2. Anomaly structure

Anomalies are at the center of SPIEL's indexing representation. They summarize what is interesting in a story, bringing together five pieces of information: viewer, perspective, agent, perspective contrast, and actual contrast. See figure 3.3.

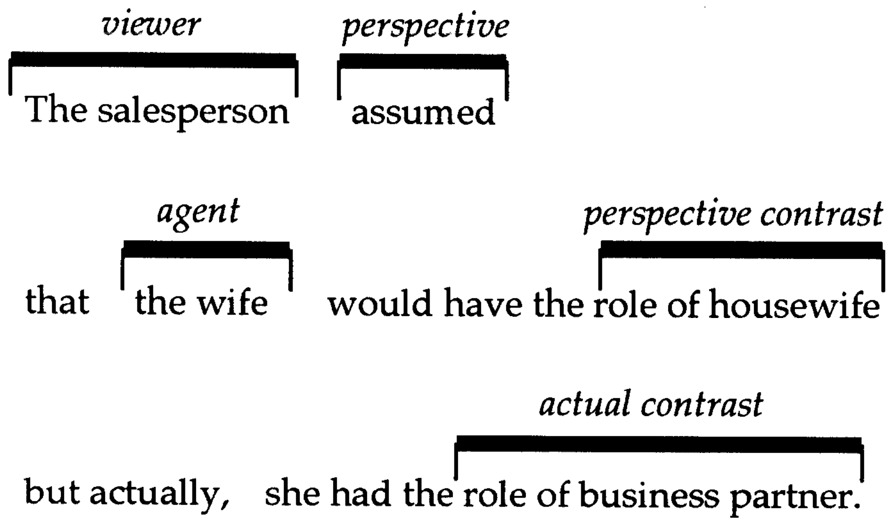

Figure 3.3. An anomaly from "Wife watching TV"

The **viewer** is the person who had the expectation that was violated. Often, this is the storyteller. The **agent** is the individual that the anomalous expectation was directed towards. This may be the same as the viewer, as in, "I thought I would fail, but I succeeded." For SPIEL's purposes, all expectations are about social actors, construed broadly to include social groups like companies.

The **perspective** is one of the five types of anomaly discussed above: assumed, wanted, feared, standard, or perceived. The **perspective contrast** is the expectation that the viewer had about the agent. The anomaly places it in opposition to the **actual contrast**, which is what the story reports as actually occurring.

3.5.3. Intentional chains

Anomalies are good summaries of what is interesting about a story, but they do not explain why the anomalous events occurred. To be able to tell its stories at times when they will help the student build explanations, SPIEL needs to have explanations for its stories. Consider the following situation. SPIEL is trying to advise a student who is having trouble with a gatekeeper, an intermediary who controls access to a decision-making client. There are many stories with a relevant anomaly:

> The salesperson wants the gatekeeper to allow access to the decision-maker, but the gatekeeper does not co-operate.

One of these stories might be relevant, but which one? Knowing no more than the anomalies, the storyteller could do no better than random guesswork.

Suppose we know that in Story A, the gatekeeper is the office manager at a law firm, and in Story B, the gatekeeper is the spouse of the owner of a small hardware store. If the student were selling to a law firm, we would probably want to prefer Story A, since the difference in *social context* between the selling situations may make a significant difference for the way the task of selling is performed.

Let us further imagine that in Story A, the gatekeeper's goal is to protect the partners from interruption since their time is considered valuable, and in Story B, the gatekeeper is interested in art and graphic design, and wants to be involved in the layout of the ad. And suppose that, in the student's situation, the gatekeeper is a closet artist who wants to have artistic input on the

	Expected
Theme:	housewife
Goal:	hospitality
Plan:	small talk
Result:	hospitality achieved
Side+:	
Side-:	

	Actual
Theme:	business-partner
Goal:	contribute to buying decision
Plan:	evaluate sales presentation
Result:	make contribution
Side+:	sale for salesperson
Side-:	

Figure 3.4. Intentional chains for "Wife watching TV"

ad. Now, Story B starts to look like the better choice, since it shows an agent with *intentions* similar to those of the simulated character the student is facing.

Settings and intentions are important for remindings because they are important parts of explanations for social actions. What makes a story relevant to a student is the relationship between the causal structure of the situation the student is in, and the causal structure of the situation described in the story. This relationship can take many forms, as I show in the chapters on storytelling strategies, but an explanation of the story is always essential.

It would be too inefficient to require a retriever to do explanation every time it considers a story. To explain the anomalies found in the stories in YELLO, SPIEL would have to have a deep understanding of social reasoning and social convention, and also knowledge of the specialized social skills involved in selling. This is why it (and the UIF) uses explanations as part of the indices themselves.

This is not a cognitively-implausible stance, since the explanations of events would have to be formed by an understander in the course of processing a story when it is told or experiencing first-hand the events from which it is made (Kolodner 1993). Since they are clearly useful as labels, it makes sense that an understander would preserve explanations for indexing purposes.

Figure 3.4 shows the two intentional chains from an index for "Wife watching TV".

> The salesperson expected the woman to play the theme of wife (theme), to have the corresponding goal of contributing socially to the sales call (goal), to carry on small talk (plan), resulting in a pleasant meeting (result).
>
> Actually, the woman had the theme of business partner (theme), had the goal of contributing to the decision (goal), evaluated the salesperson's presentation (plan), resulting in contribution to the decision (result), with side-effect of a successful sale for the salesperson (side+).

3.5.4. Settings

Social actions always take place against a background, or social setting. Every social setting has its own rules of behavior. This is true of common everyday settings: think of the difference between acceptable behavior at a church social and a rock-n-roll club, but it is also true of the specialized social settings in which the skills taught by GuSS are practiced. Selling a Yellow Pages ad to a tattoo artist is different than selling the same thing to a corporate law

Scene:	pre-call
Business type:	service
Business size:	small

Figure 3.5. Physical setting for "Wife watching TV"

firm. The same kinds of techniques may be employed in these settings and many of the same principles apply, but the fine points of social performance are different.

The UIF incorporated the notion of setting, but paid it little attention. It is difficult to say anything general about settings because what background is important depends on the story itself. Again, the notion of explanation is important. Is the fact that a client is a lawyer an important piece of information in a story about selling Yellow Pages advertising? It is, if the salesperson uses a strategy that is geared toward law firms, or if the client has needs specific to lawyers that the story addresses.

In SPIEL, settings are treated as possible contributors to the explanations in the index. I have found that there are three types of information that typically figure in supporting explanations about selling Yellow Pages advertising. These are the stage in the sales process that the story describes, the type of business the client in engaged in and the size of that business. Figure 3.5 shows the setting for "Wife watching TV," including these three slots.

The stage of the sales process is often important because each stage establishes pre-conditions for the next. The information-gathering phase (called the "pre-call") lets the salesperson gather the knowledge necessary to prepare a sales presentation. A plan to ask about a client's competitors during the pre-call serves the goal of persuading the customer by enabling arguments about competition that can be made later, for example. The type and size of the client's business often figures in a salesperson's approach. For example, a plan to sell a small ad may be related to the small size of the client's business. A plan to build an elaborate presentation may be explained by the fact that the customer is a lawyer, and professionals, such as doctors and lawyers, often require such special treatment.

The relationships between agents, as represented in the UIF in the context grid, have a role similar to the social setting as possible contributors to the story's explanation. For example, the fact that someone is the spouse of a client may help explain why the salesperson had a certain expectation about that person. Because these two types of information are treated similarly, I have done away with the term "context grid" and introduced a notion of setting that has two parts: a social setting, the relationships between agents (formerly context grid); and a physical setting, the other aspects of the story's context (formerly global setting slot).

Instead of the grid of two-place relations in the UIF, the relationships between actors in a story are represented as a list of relations between individuals that can be of any order. In the index of "Wife watching TV", there is an instance of the seller/sales target relationship, a business partnership, and a marriage. SPIEL uses this information to align the actors in the story with the characters the student encounters in the simulated world.

We are now ready to put all of the parts of an index together. Figure 3.6 is the entire "Wife watching TV" index that I have described in this section, in a UIF-style format.

As Domeshek also found, it is important that the social setting represent relationships as structure; the simple list of the types of relationships shown in the UIF-style index is insufficient. In SPIEL's indices, each entry is a memory organization packet (MOP) (Schank 1982) that packages other pieces of representation. Figure 3.7 shows how the actors in "Wife watching TV" are packaged by the associated social context.

Physical setting:	pre-call, service-business, small
Social setting:	business-partner, seller, married-couple
Viewer:	salesperson
Perspective:	expected
Agent:	wife
Anomaly Type:	theme

	Expected	*Actual*
Theme:	housewife	business-partner
Goal:	hospitality	contribute to buying decision
Plan:	small-talk	evaluate sales presentation
Result:		serve contribution
Side+:		sale for salesperson
Side-		

Figure 3.6. Index for "Wife watching TV"

Because each term found in the index is a MOP, there is also structure in the other parts of the index that is hidden in the UIF format. Figure 3.8 shows the same index including the structure of its MOPs. In the structured version, it is easy to see that any piece of the index can be extracted and used in reasoning without losing its connection to the rest. The instance of housewife theme has "woman" as the actor, the same person who has the marriage relationship to the auto glass man in the social setting, etc. Although this representation is a truer picture of the index, I use the UIF-style format for simplicity.

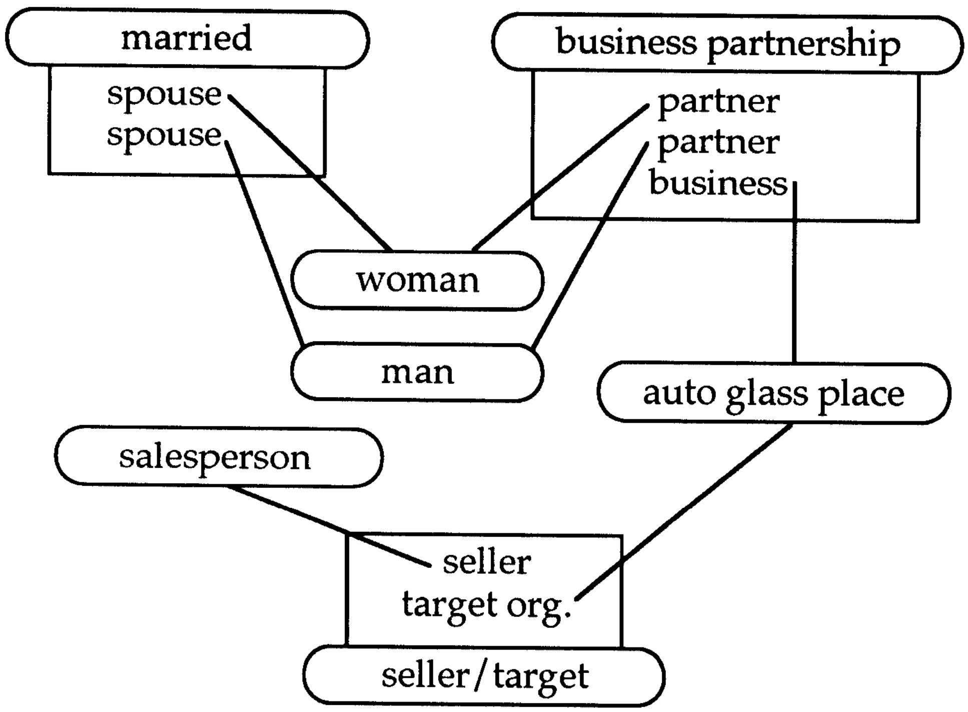

Figure 3.7. Actors and social setting in "Wife watching TV"

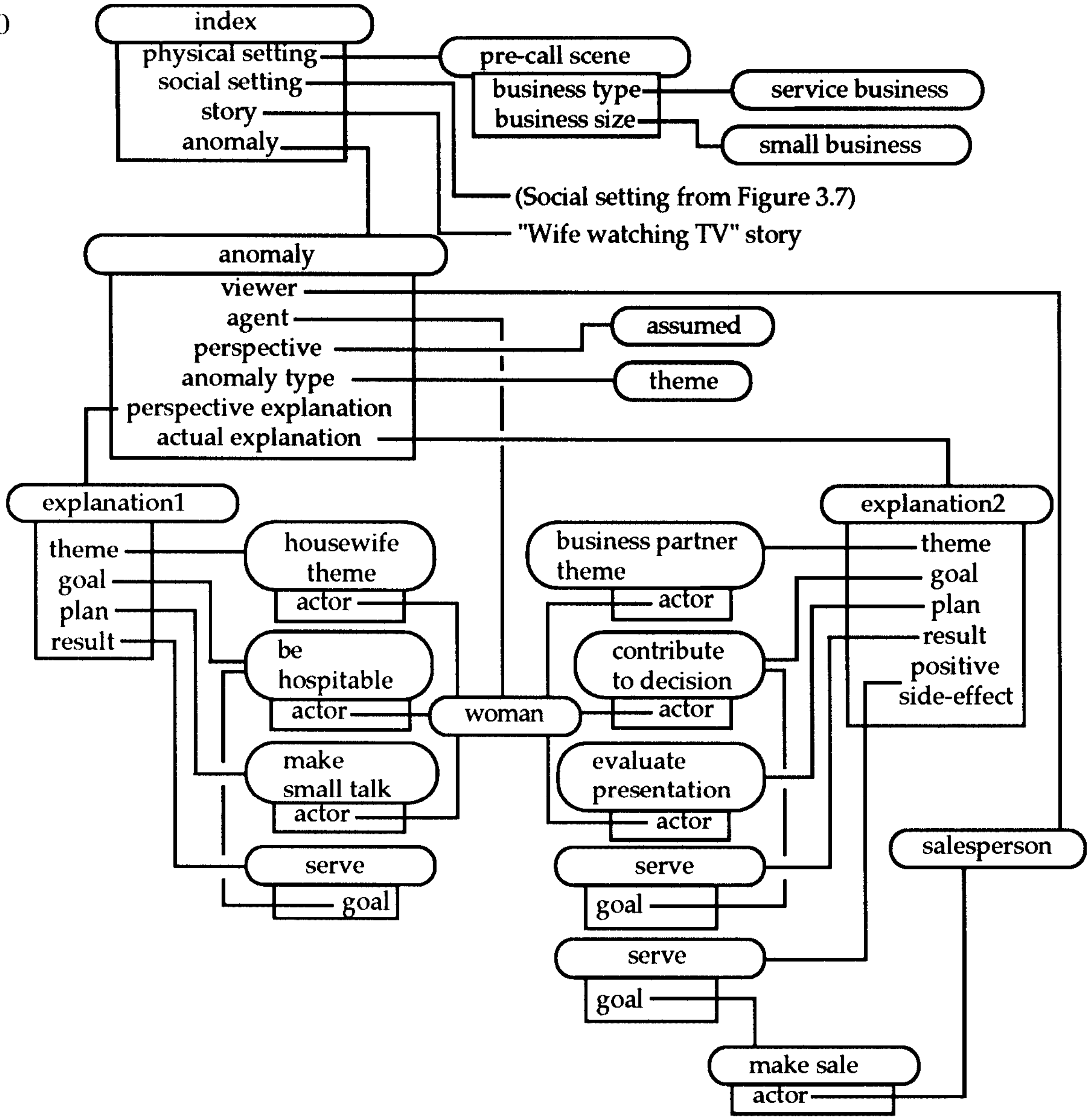

Figure 3.8. Full index for "Wife watching TV" showing internal structure.

3.6. SPIEL's indexing vocabulary

The UIF did more than show how indices should be organized, although that is probably its most significant contribution. Along with the indexing frame, the UIF theory also established the classes of fillers for each part of the frame and some representation of these classes. It did not aim for a complete representational theory, intending to be suggestive rather than exhaustive. The vocabulary proposed along with the UIF has many interesting facets, however, and I have incorporated them into SPIEL's indexing language where they have been useful.

Like the index structure, the indexing vocabulary is also a function of the task served by retrieval. The UIF work and others on the indexing problem have considered cross-domain retrieval to be particularly important: the capacity for a system to get reminded of how to fix an airplane while working on a term paper (McDougal, Hammond and Seifert 1991), for example. This is not SPIEL's mandate. SPIEL presents stories about a domain to students who are learning about the same domain. It uses domain-dependent criteria as much as it can.

To cite just one example, in selling YELLO pages ads, one persuasive technique is to show a "keyed ad study." A keyed ad study is a case study in which a customer is given a special phone number that is placed only in their Yellow Pages ad. The volume of calls received at that number is a direct reflection of how many customers are using the Yellow Pages as a source.

There are of course similar types of activity in other domains, other ways to convince a customer that the item being sold is in fact worth its cost by appealing to the experience of other satisfied customers measured in a quantitative way. If "keyed ad study" is represented in terms of this abstraction, it will be more useful for cross-domain analogy, but it will not be any more useful to SPIEL. The students of YELLO will need to decide when to use a keyed ad study, as opposed to some other persuasive tool, and reference to domain-dependent criteria will best address this question.

Again it is useful to compare SPIEL with ABBY, whose indices support a similar function. ABBY gives advice by retrieving stories about social behavior, in particular family and love relationships. It did not have to do cross-domain retrieval: to retrieve stories about marriage in response to situations involving accounting procedures, for example. Its indices contained features at different levels of abstraction but it used them solely for within-domain comparison.

A difference that renders comparison somewhat difficult is that ABBY performed retrieval through direct comparison of a retrieval probe to indices. SPIEL transforms its indices and translates them into the language of the GuSS simulation, which is where recognition and retrieval occur. In other words, where ABBY (and most case retrievers) have labels that support retrieval directly, SPIEL needs indices that support inferences about recognition conditions. Also, as we have seen, some of the distinctions that Domeshek was able to draw turn out to be less important for SPIEL since they cannot be easily recognized at retrieval time.

ABBY also had a much larger and more varied range of experience than is found among those performing the same job in the same company as is the case for the stories in SPIEL. ABBY had to incorporate many more distinctions in its vocabulary. The representational vocabulary in SPIEL is correspondingly more modest, both in the detail of its fine structure and its coverage.

A final difference between the systems has to do with the overall aims of the research. SPIEL's indices are intended to be created by users who are labeling large corpuses of stories, users who are not necessarily computer scientists. (See the discussion at the end of this chapter on the Story Indexing Tool, which facilitates this process.) The aim was not to build a complete domain theory of selling. Its representations are closer in spirit to semi-formal representations (Lemke and Fischer 1990), intended to help user and machine communicate.

SPIEL's indexing vocabulary can be extended by the user, something that was not possible in ABBY.

SPIEL's indexing vocabulary has domain-independent and domain-dependent parts. The domain-dependent vocabulary, which is user-extensible, includes concepts that are particular to the domain of YELLO, such as the goal to sell an ad or the plan of presenting a keyed ad study as a persuasive tool. Indices are made up of these domain-dependent concepts, since they are what stories are about. Underlying the domain-dependent terms is the domain-independent vocabulary, the abstractions that SPIEL uses in reasoning about indices. It includes the basic components of the intentional chain: theme, goal, plan, and result.

3.6.1. Themes

As in the UIF, an intentional explanation always starts with a **theme**, a social role or other fairly long-lived predisposition towards certain behavior. A theme gives rise to goals that naturally co-occur. (Schank and Abelson 1977) uses the example of "luxury living." A person with this as an important theme would have goals such as acquiring the trappings of luxury and taking certain kinds of vacations.

Themes are distinguished by the threads, or areas of life, to which they relate. (Threads are discussed in [Schank et al. 1990] and [Domeshek 1992].) SPIEL uses four major threads: **economic**, **career**, **family**, and **personal**. For example, a businessperson may have an independent personality that causes him to want to avoid reliance on others' expertise. This is a personal theme, something that is particularly resistant to change, as opposed to a career theme that will change as a person's career develops.

3.6.2. Relationships

Like themes, relationships are distinguished according to threads. SPIEL uses three major threads for relationships: **career**, those aspects of a person's life that have to do with vocation, such as an employer/employee relationship; **economic**, relationships that are strictly part of the sphere of commerce, such as the relationship between a buyer and a seller; and **family** relationships. Personal themes, by their nature, do not give rise to the kinds of relationships found in SPIEL's stories.

3.6.3. Goals

Themes give rise to **goals**. SPIEL recognizes two classes of goals: stative, goals to bring about particular states of the world, and policy, goals to hold certain things constant over long periods (known as A-GOALs and M-GOALs in [Schank and Abelson 1977]). A typical stative goal is a salesperson's goal to sell a full page ad to a particular client. An example of a policy goal is the desire to preserve close rapport with a client.

In the representation of goals, I have relied on the notion of resource similar to that in the UIF. A goal can be represented as an intention to affect a resource in a particular way. SPIEL represents five different kinds of impacts that an actor may wish to have on a resource. An actor may wish to **obtain** a resource or **expend** it. Other resource goals center on holding something true of a resource over time: **maximizing** the amount of resource one has, **conserving** a resource, or attempting to **maintain** a certain level of the resource. Goals can be compared by comparing their resource impacts. For example, a plan to obtain rapport with the customer is opposite from a plan to break off (expend) relations because they have an opposite impact on the resource.

Sub-categories within these goal categories are distinguished by the type of resource under consideration. The actions that surround resources often depend on three important variables: **maintainability**, the amount of effort or other resources required to keep a resource usable; **fluidity**, the degree to which a resource can be converted into other resources; and

creatability, the uniqueness or individuality of a resource. SPIEL uses a simple two-valued combination of these features to describe any given resource. For example, money has low maintenance requirements, high fluidity, and is creatable. Using permutations of these values for the three dimensions gives eight types of resources. The second domain-independent level of the goal hierarchy is therefore made up of all of the resource effects crossed with the eight types of resources: giving 40 categories of goals.

3.6.4. Plans

To bring about a goal, an actor executes a **plan**, a set of actions designed to achieve it. A typical plan for getting a customer to believe that a certain ad will bring increased revenue is the plan to show the customer a testimonial letter from a client who has experienced good results for a Yellow Pages ad. Usually, the entire plan for accomplishing a high level goal like "sell an ad" is long and complex, but there are useful subparts that act in service of the goal and can be extracted. I use the word "plan" to refer both to entire plans whose successful completion would achieve a goal, and sub-plans that are enacted in service of a goal.

Plans form the most complex of SPIEL's categories, containing about 60 domain-independent categories. Many plans belong to more than one category. I will not cover all of them here. The first breakdown of plans has to do with a similar distinction as was made for goals. Is the plan an implementation of a policy, a **policy plan** or it is aimed towards achieving some state, an **achievement plan**?

Policy plans are those designed to control resources. This can be control over how a resource is used, a **deployment plan**; control over which resource to employ, a **selection plan**; or control over upkeep of a resource, a **maintenance plan**. For example, a salesperson might have a plan to wait until the customer has been told the price of an ad before presenting a testimonial letter attesting to its value. This is a deployment plan in which the timing of the use of a resource is of primary importance.

Achievement plans aim to create a certain state of a resource. An agent can obtain a resource through a **resource creation plan** or shed resources with a **resource expenditure** plan. Resources that are sharable, like information, can be hidden from others with a **resource protection plan**, or made available via a **resource sharing plan**. For example, a salesperson using a sharing plan might give customers marketing information.

Plans can be further distinguished by the class of resource they involve: physical, social, or mental, and by the resources used in achieving that aim. One important class of plans in YELLO are persuasion plans: creation plans for creating sales contracts. A persuasion plan that requires investment of time and effort (such as preparing and giving a presentation) is different from one in which a resource is given away in exchange (a bribe). Other persuasion plans may involve altering the contract, such as discounting a price, to make it more attractive.

3.6.5. Results

Executing a plan leads to a **result**, an impact on a goal. SPIEL recognizes five kinds of impact: two positive, two negative, and one neutral. In the realm of positive impacts, there is **achieve**. The result of achievement holds if the goal is completely achieved through the use of a plan. Less definitive, but still positive is a **serve** result. An action can contribute towards the achievement of a goal without completely achieving it, serving the goal. The negative outcomes are complementary. A **block** outcome means that the goal has been rendered unachievable by the original plan. **Threaten** means that its achievement has been hindered but not stopped. The **neutral** outcome is present when there is no impact on the goal. It is often found contrasting other results in an index. For example, a salesperson may hope that a sales pitch of a certain type will persuade the customer, but in fact it has no effect.

Table 3.1. Summary of SPIEL's indexing representation

Category	Discriminating feature	Major distinctions	Examples
Perspectives	Expectation type	assumed	assumption about someone's role
		feared	fear of customer's reaction
		hoped	hope of achieving a good result
		standard	standard of business conduct
		perceived	perception of customer reaction
Themes	Thread	economic	buyer, seller
		career	employee
		family	parent
		personal	independent living
Relationships	Thread	economic	sales target
		career	employer/employee
		family	marriage, father/son
Goals	Resource impact	obtain	get sales commission
		expend	waste time
		maximize	have the largest market share
		conserve	preserve personal status
		maintain	keep competitive advantage
Plans	Resource action	resource deployment	give testimonial after telling price
		resource selection	prefer radio ads over Yellow Pages
		resource maintenance	engage in small talk
		resource creation	buy a Yellow Pages ad
		resource expenditure	keep salesperson waiting
		resource protection	avoid giving business information
		resource sharing	give client marketing information
Results	Goal impact	achieve	making a sale
		block	failing to increase an ad program
		serve	getting the customer to agree to a point (when trying to make a sale)
		threaten	having the customer disagree about a point

3.6.6. Summary

We have now seen the six concepts that are the primary elements in SPIEL's indices: perspectives, themes, relationships, goals, plans and results. To create an index, a user interprets a story in terms of expectation failure and encodes the expectation and its underlying explanation in this vocabulary. Table 3.1 summarizes these concepts and the major distinctions that SPIEL's indexing representation uses.

3.7. The Story Indexing Tool

One important aspect of indices yet to be discussed is their generation. In many case-based problem-solving systems, new cases are added when the system has experiences. Indices for these new cases are generated automatically in the course of problem-solving. The nature of

SPIEL's case and its task precludes the system from being able to acquire cases by interpreting stories. To understand videotaped stories, it would need not just natural language understanding, but voice and possibly gesture recognition as well. Indices must be entered manually.

Tools for the efficient manual creation of indices are crucial for the development of large story bases (Ferguson et al. 1992). An indexing tool for SPIEL must do five things:

1. Allow the creation of stories so that they can be perused and labeled.
2. Enforce the structure of the index.
3. Render indices intelligible and manipulatable.
4. Allow the maintenance of an indexing vocabulary.
5. Interface with the strategy application and rule generation components of SPIEL.

Development has concentrated mainly on 2 and 3. The tool's interface to the other parts of SPIEL's processing, in particular, is not yet fully developed.

Figure 3.9 shows the screen that the Story Indexing Tool (SIT) first presents to the user. This screen lists all of the stories that have been collected for a particular GuSS application and the formats in which they are available. The "Wife watching TV" story is selected and it can be seen that both a Quicktime™ and a text version of the story are available. A brief summary of the story can be seen at the bottom of the screen. The story and its associated indices can be inspected in the indexing screen shown in figure 3.10.

The top part of the screen holds information about the story itself: the teller, the original videotape source, a link to a Quicktime™ movie, a Laserdisk segment or a text transcription. In the bottom half of the screen, the user begins constructing the index, starting with the physical and social settings, which are the same for all indices for a given story. On this screen the user has listed three actors from the story and established relationships between them, using a mechanism for browsing and instantiating SPIEL's MOPs. Figure 3.11 shows the MOP browser being used to instantiate the marriage relationship between the woman, the "Soap opera watcher," and her husband, the "Auto glass guy." (The names are solely for the user's benefit.)

The list near the bottom of the story screen shows what anomalies are associated with the story. Anomalies and their associated intentional chains are created for a story in the anomaly-creation screen, shown in figure 3.12. Here the user first fills the viewer, perspective, and agent slots to begin the English-like anomaly description: "Amber assumed that 'Soap opera watcher' would be..." Then the browser at right similar to that seen in figure 3.11 is used to select the perspective and actual contrasts, completing the description: "acting according to the theme of 'housewife-theme', but instead 'Soap opera watcher' was acting according to the theme of 'business-partner-theme,'" as underlined in the figure. The rest of the intentional chains can be similarly instantiated to complete the index.

The Story Indexing Tool was used to index all of the stories in YELLO's story base, 178 in total. Although a complex index has from 20-30 MOPs, an experienced user can create one in several minutes with the tool. Development of the tool is continuing as it is being applied to index stories in other GuSS applications.

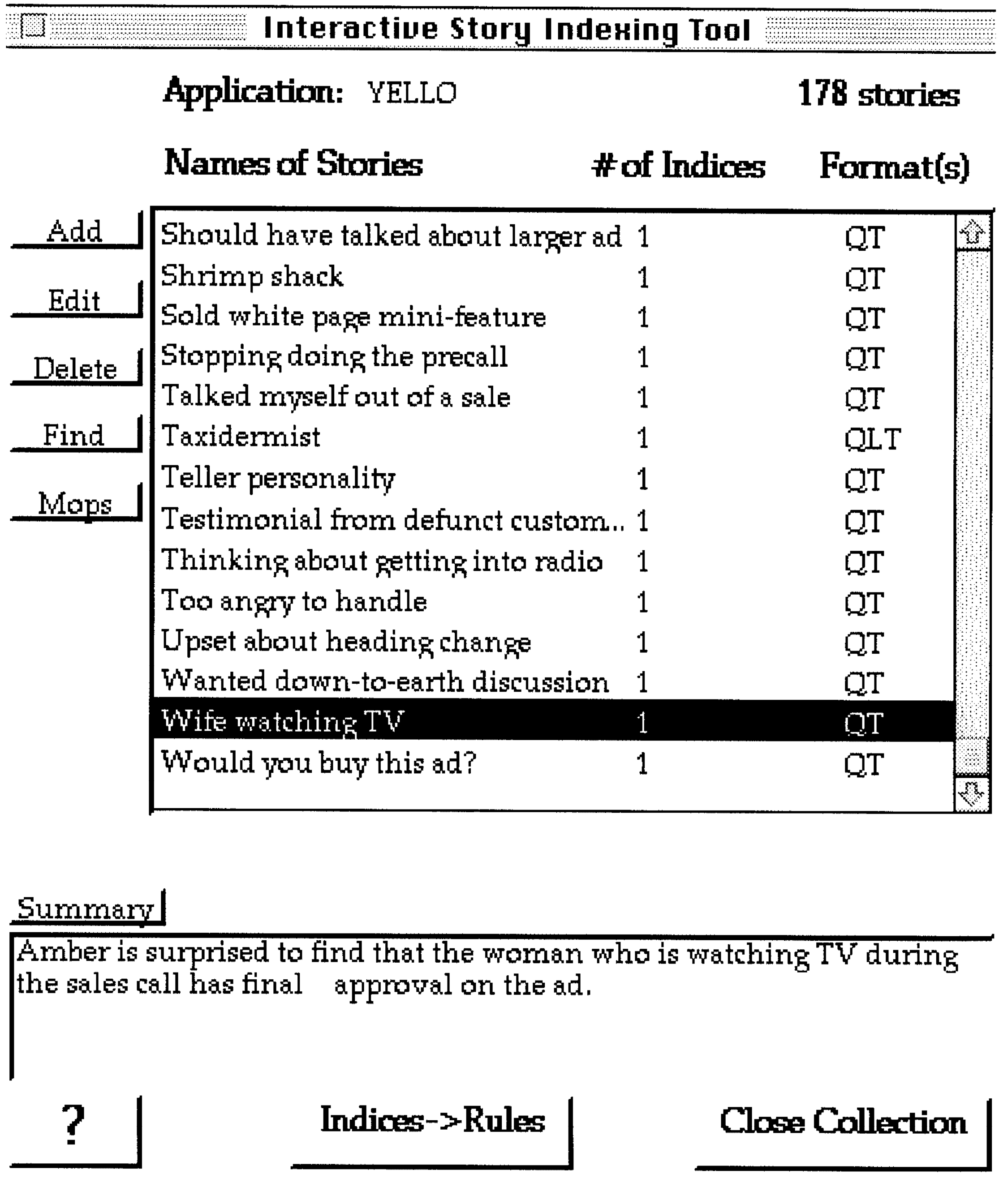

Figure 3.9. Story Indexing Tool: Listing of stories

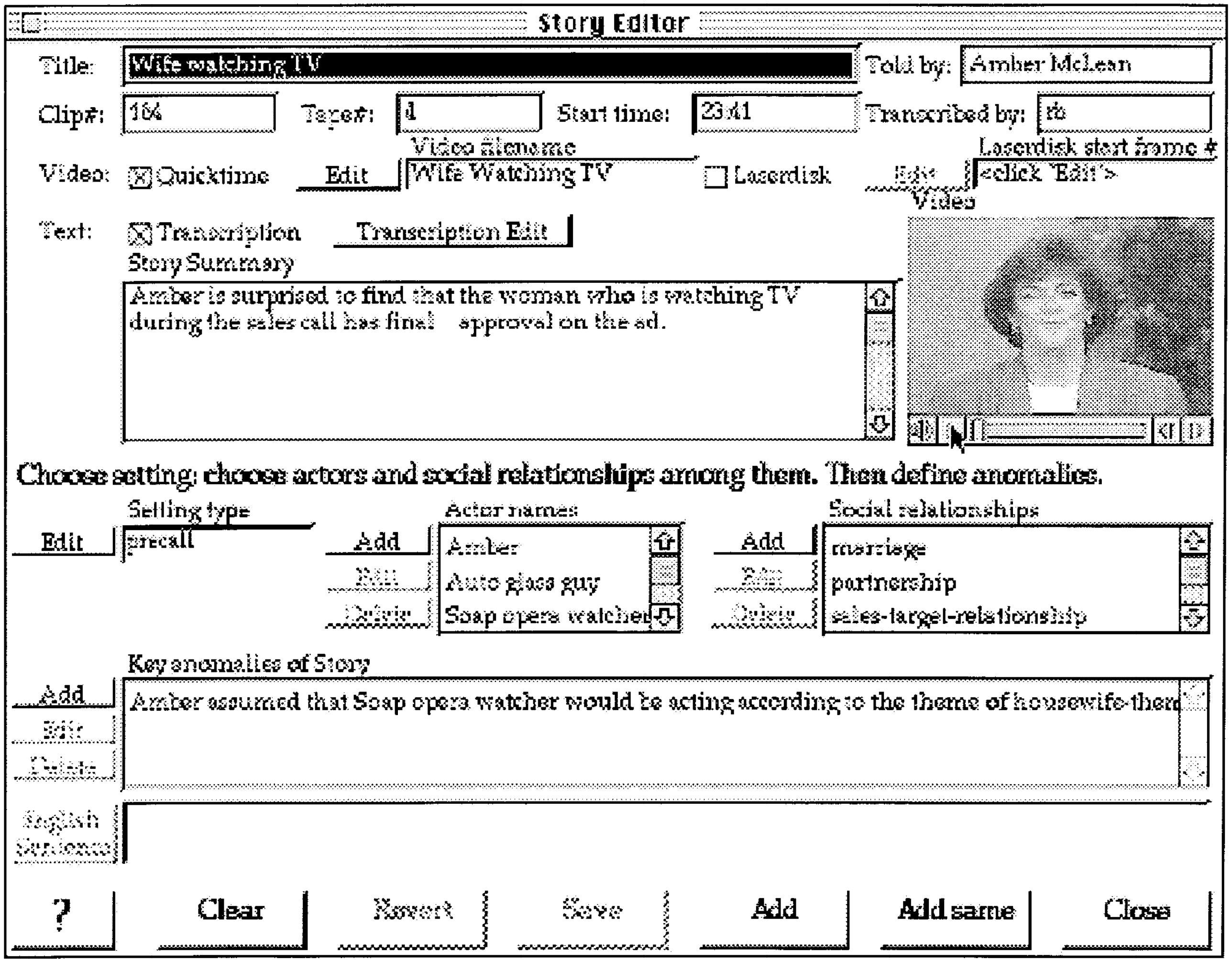

Figure 3.10. Story Indexing Tool: Story infromation

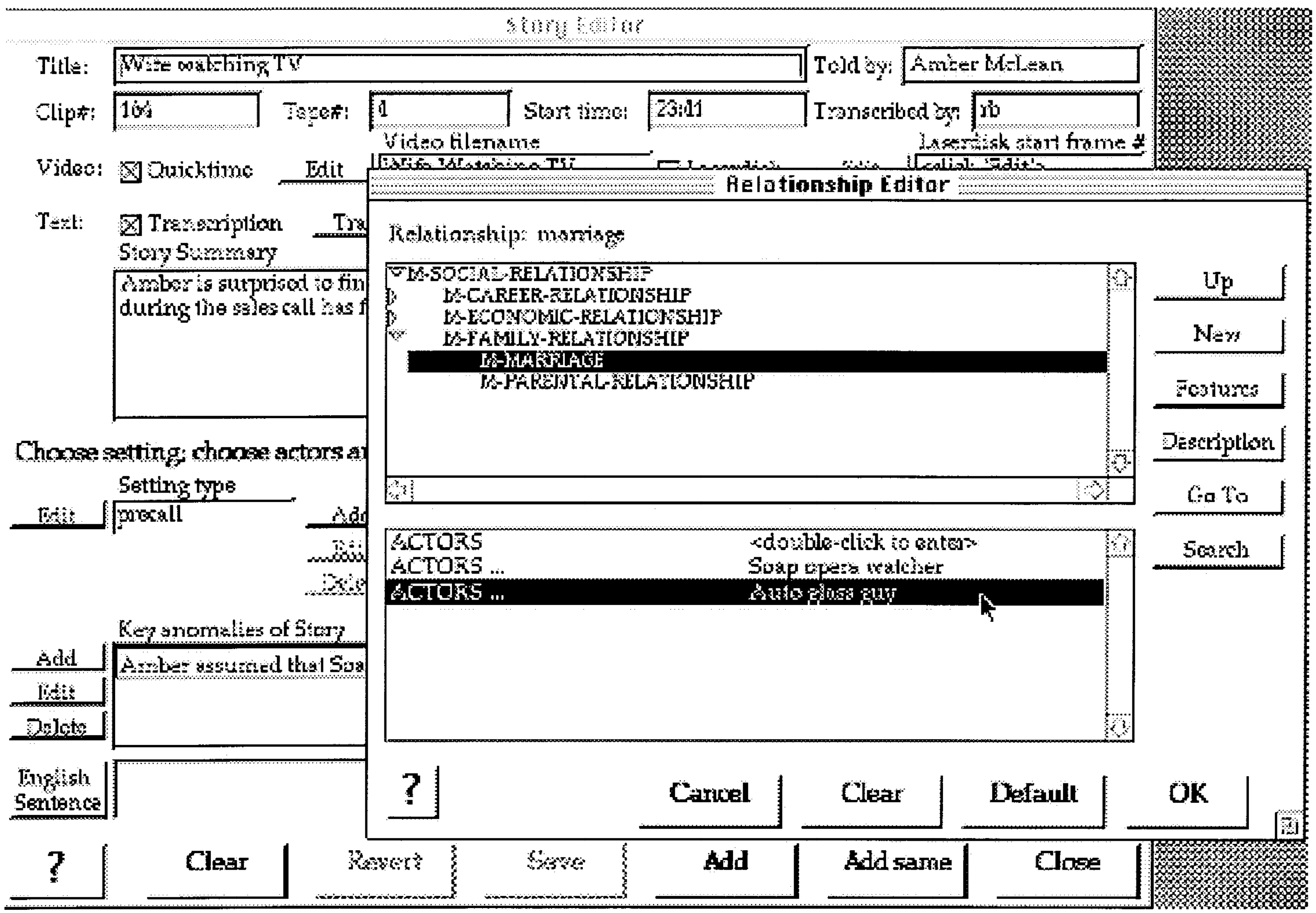

Figure 3.11. Story Indexing Tool: Instantiating a social relationship

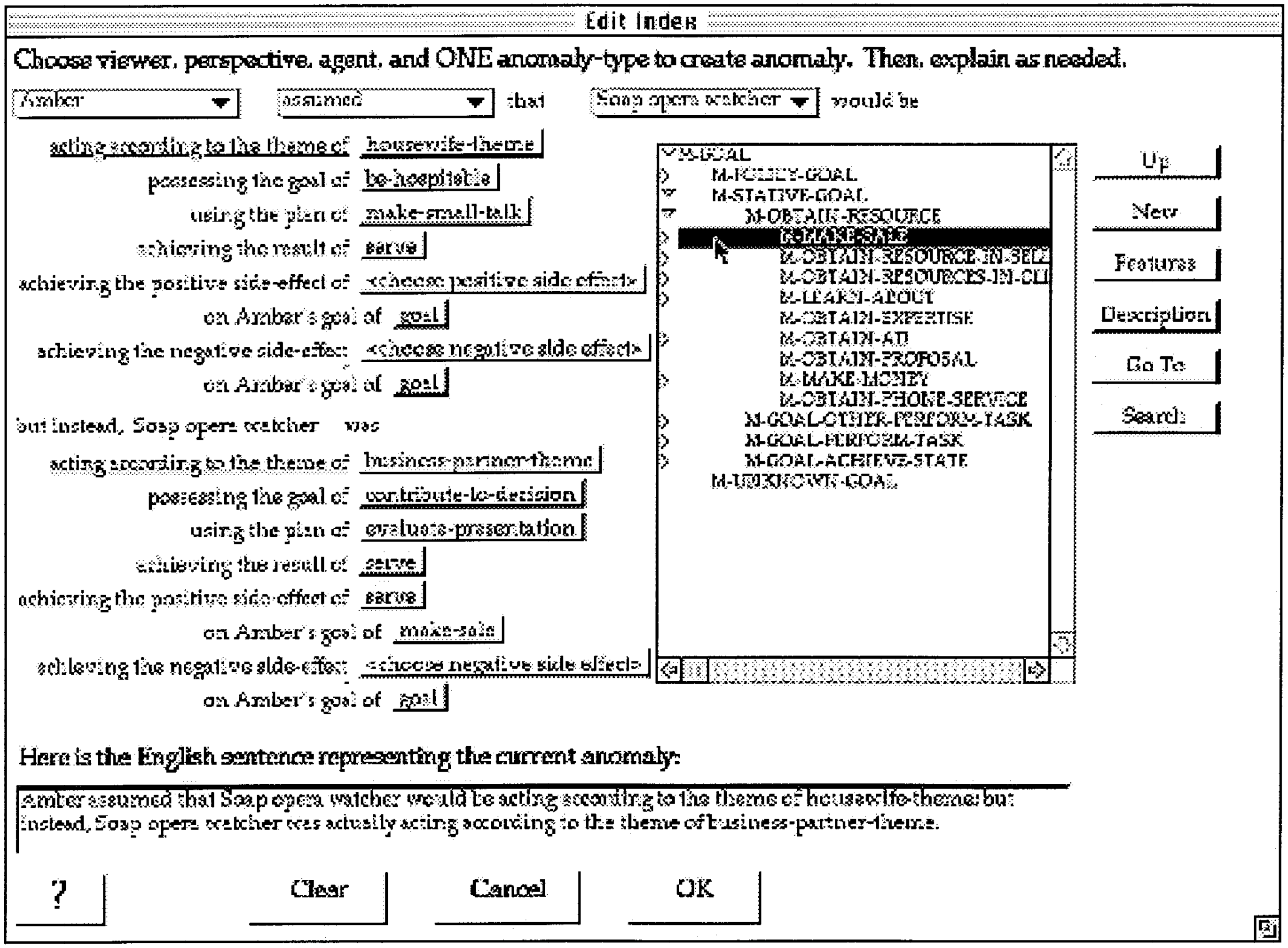

Figure 3.12. Story Indexing Tool: Creating anomaly and associated explanations

3.8. Conclusion

An indexing representation is a compromise. It must balance the need for predictively-powerful features that will recall useful stories against the realization that such features are often difficult to discover at retrieval time. SPIEL is modeled after the Universal Indexing Frame (Schank et al. 1990) in its selection of features that best predict the usefulness of a story. SPIEL uses a simplified version of the UIF, tailored for the needs of a tutorial storyteller. An index is represented by an anomalous occurrence that marks the story as interesting, which is supported by explanations of how that anomaly arose, and situated in a social and physical setting.

I also drew inspiration from ABBY's indexing theory (Domeshek 1992), but found the retrieval constraints imposed by the GuSS architecture and the YELLO application to be significant. Some of the causal determinants that Domeshek found useful in ABBY have turned out to be too difficult for SPIEL to detect at retrieval time, since they would require deep modeling of student intentions.

SPIEL's indexing representation proved sufficient to represent 178 stories in the YELLO corpus. The anomaly has proved to be a natural format for users to think about the point of a story. An earlier version using a very similar vocabulary, at least in its domain-independent levels, was used to index a similar number of stories about selling consulting services. SPIEL's domain-independent vocabulary and index structure are now being employed in the development of a GuSS application in the area of project management. These examples argue well for the system's sufficiency for representation, and the Story Indexing Tool has made these indices easy to construct, despite their internal complexity.

However, the ultimate test of a representation is its sufficiency for a task. SPIEL's indices should be judged by how well they support the system's storage and retrieval processing. The next chapters discuss SPIEL's processing from the application of the storytelling strategies to the generation of rules. I will show in these chapters how the indexing vocabulary supports the inferences required for retrieval.

4. Telling Relevant Stories

4.1. Implementing the case-based teaching architecture

The combination of a storytelling tutor with a practice environment is known as the *case-based teaching architecture* (Schank 1990). The student works in a simulated environment, which the tutor monitors in order to present stories from its library when they are relevant to what the student is doing. For such a tutor, much hinges on the notion of relevance: what does it mean for a story to be relevant? This question is central to the case-based teaching architecture, and its answer forms the basis for SPIEL's implementation. This chapter outlines SPIEL's solution to the question of relevance.

For SPIEL, a storytelling episode creates an analogy between the experiences of a practitioner reported in a story and the experiences of a student enacted in a GuSS application. This analogy contains a certain amount of overlap and a certain degree of difference. The inspiration for the case-based teaching architecture is found in apprenticeship situations where a novice practices a skill under the close supervision of an expert. We can imagine a task such as car repair being taught in this way. Suppose a novice mechanic encounters an electric fuel pump operating erratically and starts replacing it. The experienced car mechanic watching over his shoulder might intervene with an appropriate example from his own experience:

> The fuel pump is a good guess, but you might want to eliminate other possibilities.
>
> One time I was brought a Ford pickup. The customer complained that it was stalling all the time for no apparent reason. I ran it for awhile, and found it did stall occasionally. I checked the fuel lines and found that they were OK but fuel was getting to the carburetor intermittently. Looked like a bad fuel pump, so I ordered a new one. When I got it in and replaced it, the new fuel pump acted the same. I puzzled over this for awhile, and then checked the electrical supply. It turned out there was corrosion on the fuse for the fuel pump that caused an intermittent electric supply to the pump. Replacing the fuse holder solved the problem. The old fuel pump was fine.
>
> You might want to check all possible causes of failure before you start to replace a part.

What the student's mentor does is present an analogous situation through the story. The novice can compare the situation in the story with his own situation, as shown in figure 4.1. Most aspects of the story are very similar. The only difference is that the student has not completed the repair and therefore has not verified his hypothesis of the fault, whereas the story contains a description of an actual fault that is different from the hypothesis. It is this difference that makes the story relevant. The student would be naturally drawn to consider whether his case is likely to end the same way as the one described in the story. Since acting on a faulty hypothesis wastes time, the student may want to rule out the hypothesis that the story suggests (and other possible causes of the malfunction) before going further.

Both similarity and difference are therefore important in the relevance equation. The analogy created by the story's telling must contain enough similarity so that a student will accept it as being a good analogy. A completely analogous story is not that instructive, however. If the story in the example were about a fuel pump that were faulty, just as the student thinks, it would not be very interesting. Students engaged in learning-by-doing will learn best about the situation they are in by acting in it. A story that has no differences from their experience does not add to the learning-by-doing experience.

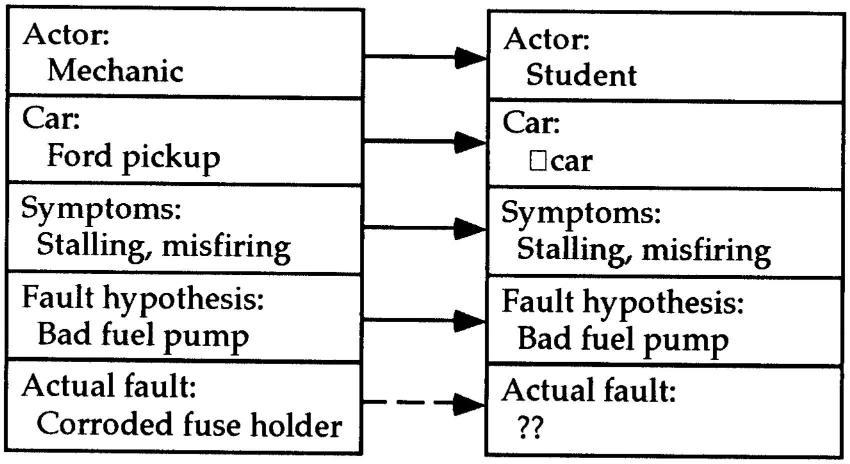

Figure 4.1. An example analogy formed by storytelling

A story that has some important difference, on the other hand, challenges the student's understanding of the situation. In the example, the story challenges the student's explanation of the fault, by showing a time when that explanation did not work. Another kind of story might address the typicality of the student's situation: showing failure where the student has achieved success. These stories raise explanation questions (Schank 1986) that drive the student towards a deeper understanding of the subject matter. Without such challenges, students will tend to leave important assumptions untested (Klahr, Dunbar and Fay 1990) and tend to see only what confirms their hypotheses (Schauble et al. 1991).

To assess the relevance of a story, a storyteller must be capable of recognizing both similarities and differences between stories and the situations in which they are told. To assess similarity and difference, there are three kinds of comparisons that a tutor can make in the case-based teaching architecture: comparisons based on what the student **does** in the task environment, comparisons based on what the student **sees** in the task environment, and comparisons based on inferences about the student's **reasoning** about the task environment.

I have identified 13 classes of relevance that stories can have in a case-based teaching architecture. SPIEL finds stories that satisfy these conditions of relevance by using ***storytelling strategies*** that define opportune moments for story presentation. In the car repair example, the kind of relevance that is sought is similarity of hypothesis and difference between the hypothesis and the actual fault. The strategy at work might be called "Propose alternative hypothesis." Each combination of a story with a storytelling strategy is a ***tutorial opportunity***, a chance for the system to tell a relevant story in an educational way. Consider the storytelling situation that arises in example 4.1 (one of the examples from the dialog in Chapter 2). The tutorial opportunity here is that the student appears to be making an assumption about Mrs. Swain, an assumption similar to the one that turns out to be violated, in the story.

4.1.1. An architecture for finding opportunities to tell stories

To find tutorial opportunities, a storyteller needs two kinds of knowledge: the knowledge of what a given story is about and the knowledge of how, in general, stories can make educational points. The first kind of knowledge comes from the indices of stories discussed in the previous chapter. The second is an abstract body of knowledge about tutorial storytelling. Note that all of this knowledge is available to SPIEL before the student sits down to use the GuSS simulation. This fact suggests a particular kind of architecture for recognition, one where tutorial opportunities are characterized ahead of time and then searched for during the student's interaction with the system.

- Student arrives for a pre-call appointment with a roofing contractor.
- The contractor's wife greets her and they carry on small talk.
- The student turns immediately to the contractor when he arrives.

A warning about something you just did

If you assume that Mrs. Swain will not have a role in the business, you may be surprised. Here is a story in which an account executive had a similar assumption that did not hold:

I went to this auto glass place one time where I had the biggest surprise. I walked in; it was big, burly man; he talked about auto glass. So we were working on a display ad for him.

It was kind of a rinky-dink shop and there was a TV playing and a lady there watching the TV. It was a soap opera in the afternoon. I talked to the man a lot but yet the woman seemed to be listening, she was asking a couple of questions. She talked about the soap opera a little bit and about the weather.

It turns out that after he and I worked on the ad, he gave it to her to approve. It turns out that after I brought it back to approve, she approved the actual dollar amount. He was there to tell me about the business, but his wife was there to hand over the check.

So if I had ignored her or had not given her the time of day or the respect that she was deserved, I wouldn't have made that sale. It's important when you walk in, to really listen to everyone and to really pay attention to whatever is going on that you see.

An assumption that a spouse will not have a role in the business may be unrealistic.

Example 4.1. Telling the "Wife watching TV" story.

SPIEL's processing is divided into two phases: storage time and retrieval time. In the first phase, storage, new stories are entered into the system and the system considers what tutorial opportunities it should look for in order to tell them. During retrieval, a student interacts with the GuSS system and SPIEL evaluates, in parallel, the features linked to tutorial opportunities, retrieving and presenting stories when their associated opportunities are recognized. In this chapter, I will concentrate on the storage-time aspects of SPIEL, and in particular on the role of storytelling strategies in the characterization of tutorial opportunities. Rule generation and run-time application are covered in Chapter 9.

Figure 4.2 shows the steps that occur in each of these phases. At storage time,

1. Indices are attached to stories manually using an indexing tool.
2. The storytelling strategies are applied to each index. For every index that a strategy finds compatible, it generates a recognition condition description (RCD). The RCD describes a tutorial opportunity afforded by the story: a situation in which the story could be told using the strategy.
3. The rule generator then processes each RCD and generates a set of recognition rules for the tutorial opportunity described.

At retrieval time,

1. The student interacts with the GuSS simulation. The recognition rules are matched against the state of the simulation as the activity proceeds.
2. When a tutorial opportunity is recognized, the story associated with that opportunity is retrieved.
3. The retrieval conditions and the strategy are fed into a language generator to create explanations of the reminding for the student's benefit. The generator recalls natural language templates for the headline, bridge and coda associated with the strategy and fills them in.

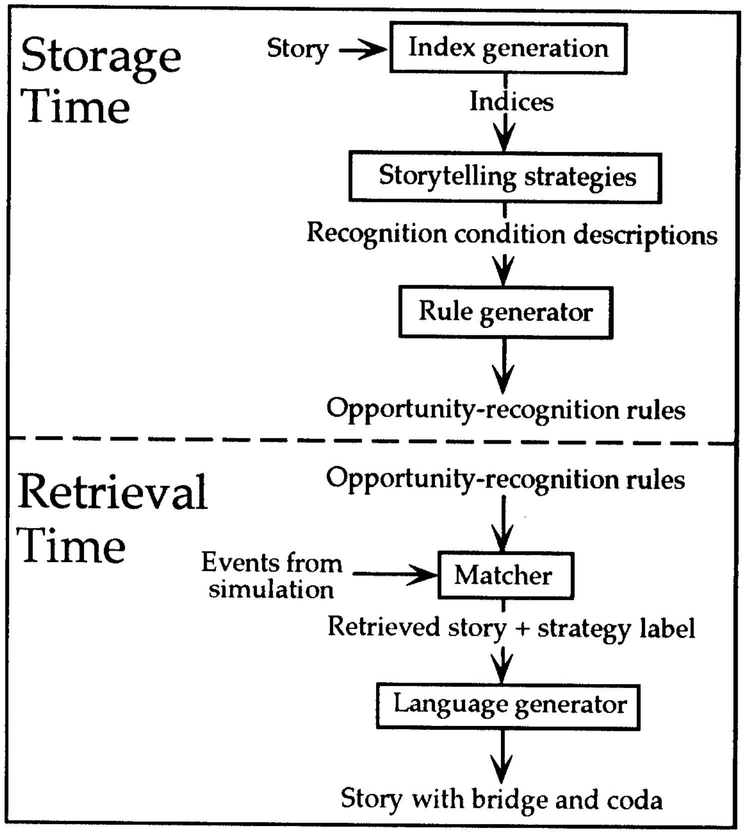

Figure 4.2. Storage and retrieval phases of SPIEL.

4.1.2. The elaborate-and-index model of opportunism

This model of opportunism is one that Birnbaum called the "elaborate and index" model of opportunism:

> "...spend some effort, when a goal is formed, to determine a number of situations in which it might be easily satisfied...and then index the goal in terms of all the features that might arise in such situations." (Birnbaum 1986, 146)

As Birnbaum notes, the elaborate-and-index model is not a general model of opportunism. Abstract goals such as "get revenge" have so many possible conditions of realization that it would be impossible for a planner to determine in advance what they all are. However, SPIEL's problem of opportunism is much simpler than the general one. SPIEL need only be concerned with those opportunities that can arise within the GuSS simulation. The range of possible realizations is limited in GuSS since the student is constrained by the program's interface and the simulation operates in known ways.

The elaborate-and-index model treats goals as independent, despite the fact that opportunistic behavior often requires trading one goal off against another. If the opportunity to pursue several goals arises, an opportunistic system has to decide which goals should be acted upon and which must be ignored. This involves making tradeoffs: it requires understanding the relative priorities of goals and their possible interactions. Storytelling does not involve such tradeoffs. If several stories are all relevant to a student's situation, a storyteller can signal the student that they are all available and provide the option to view as many or as few as desired.

Because storytelling goals do not interact, each tutorial opportunity can be evaluated independently by the elaborate-and-index method.

Birnbaum argues that in the worst case an unbounded amount of inference may be required for a planner to notice that an opportunity exists. This happens if there is a very involved connection between a goal and its realization. The need for such inference is a problem for the elaborate-and-index model. Complex opportunity recognition requires bi-directional inference from both the goal and the situation that is a possible opportunity, so such inference cannot be part of the system's initial elaboration of the goal. It is also true that the elaborate-and-index model breaks down if unbounded inference must be performed in order to recognize the presence of the features used to index goals. The whole idea of the model is that only simple tests will be required to recognize opportunities once they have been elaborated.

In the case-based teaching architecture, however, the tutor is trying to tell stories that will make sense in a very straightforward way. Distant or novel analogies will probably be more confusing than helpful. SPIEL's storytelling strategies seek to make stories relevant in simple ways, and they need only limited amounts of inference to do this.

The simulated world limits the space of actions that a student can take and the space of possible reactions within the simulation. Storytelling strategies single out precise areas in these spaces as tutorial opportunities. The storyteller does not have to concern itself with novel opportunities, and those that it must recognize do not require unbounded inference. Further, because storytelling opportunities do not interact very strongly, they can be evaluated independently of each other. The elaborate-and-index model of opportunism is therefore appropriate for the case-based teaching architecture.

4.2. The recognition condition description

How would an opportunity to tell the "Corroded fuse" story be recognized in a hypothetical car repair tutor? The tutor would look for (1) the student to be working on a car with erratic fuel pump behavior and (2) to be at the point of deciding what to repair. Then it would look for (3) the student to decide to replace the fuel pump. At this point, it would (4) tell the "corroded fuse" story as a suggestion to the student of other possible hypotheses.

A sequence of conditions for recognizing a storytelling opportunity is called in SPIEL a *recognition condition description* or RCD. The RCD describes what situation the student would have to encounter for the story to be relevant. Recognition condition descriptions have three parts: (1) a context in which the student might encounter the circumstances that make the story worth telling, (2) conditions that indicate that the story is relevant in that context, and (3) the story itself and the manner in which it should be presented. The test for the establishment of the context I call the *trigger*. The recognition of the tutorial opportunity itself is performed by tests that gather *evidence*. The storytelling action that follows is the *presentation*. When the triggering conditions are met, SPIEL will try to find evidence that a tutorial opportunity is present. If it finds the evidence, it presents the story. These three parts can be thought of as filling roles in the following abstract rule form:

WHEN *trigger*,
LOOK FOR *evidence*,
THEN *presentation*.

4.2.1. Triggering conditions

The trigger describes the context in which it makes sense to look for a tutorial opportunity. A story whose triggering conditions are met will be under active consideration until there is evidence that it does not apply. Triggers, therefore, are the first steps in SPIEL's incremental retrieval. They must be relatively stable so that the system can gather evidence while the trigger holds true.

SPIEL uses triggers derived from stories' social and physical settings. This is because the social activities SPIEL teaches are highly associated with particular social configurations. For example, selling requires the presence of a salesperson, something to sell, and a person to sell it to. Particular sales strategies are often predicated on the existence of certain social relationships, such as a partnership in a business. Settings remain stable during periods of social interaction.

An important part of the setting of a story is the part of the overall task that it describes. Usually, a tutorial opportunity will arise in the same step of the task that the story describes. In the car repair example, the story was about the hypothesis of a fault, and the opportunity to tell the story arose in the same stage of the repair process.

4.2.2. Evidence conditions

After the triggering conditions have been satisfied, a tutorial opportunity is recognized through the evidence conditions of the RCD. Storytelling strategies differ most in how they create evidence conditions. A strategy that seeks to show a student a new way to do something will look for the student doing something very different from what someone did in the story, so that the story will appear to present an alternative. A strategy that seeks to project into the future possible consequences of a student's actions will look for the student doing something very similar to what someone does in the story but without the consequences being apparent.

What the evidence conditions do not need to do is specify in detail how the tutorial opportunity might be recognized in the simulation. For example, the RCD for the "corroded fuse" story does not say whether the decision to replace the fuel pump should be recognized when the student removes of the pump or when the student places an order for a new fuel pump. Ultimately, the RCD must be operationalized, turned into conditions that can be tested against the student's actions and the state of the simulated world. As described in Chapter 2, the operational component of SPIEL is a tutorial module that uses decision rules to operate within the GuSS simulation.

This level of operational detail is provided by the rule generator that translates the RCD into simulation-specific conditions. To create the description, SPIEL need not be concerned if the simulation requires the student to place an order for a new part, to pick it off the shelf, or to select it from a computer menu. The rule generator must take concepts like "decision to replace fuel pump" and use its knowledge of the simulation to turn this into a rule that looks for specific actions with specific meanings. Because the RCD is independent of the implementation, the conditions it describes will be recognizable in any sufficiently rich simulation where the student can practice the target activity.

4.3. Applying storytelling strategies

4.3.1. Filtering for compatibility

Before a storytelling strategy can be applied to an index, SPIEL has to determine if the strategy and the story are compatible. Each strategy has a filter that can be applied to indices to determine their compatibility. For example, **Warn about plan** is a strategy whose intent is to warn the student. It looks for stories that show someone in a position similar to the student's experiencing failure, so it can present the stories if the student looks to be heading in the same wrong direction. Stories that show success would probably not make convincing warnings.

The purpose of the filter is to select stories that are capable of making a certain kind of analogy. This analogy has a slightly different character in each strategy and each filter looks for something slightly different. Almost always the filter seeks to find someone in the story who

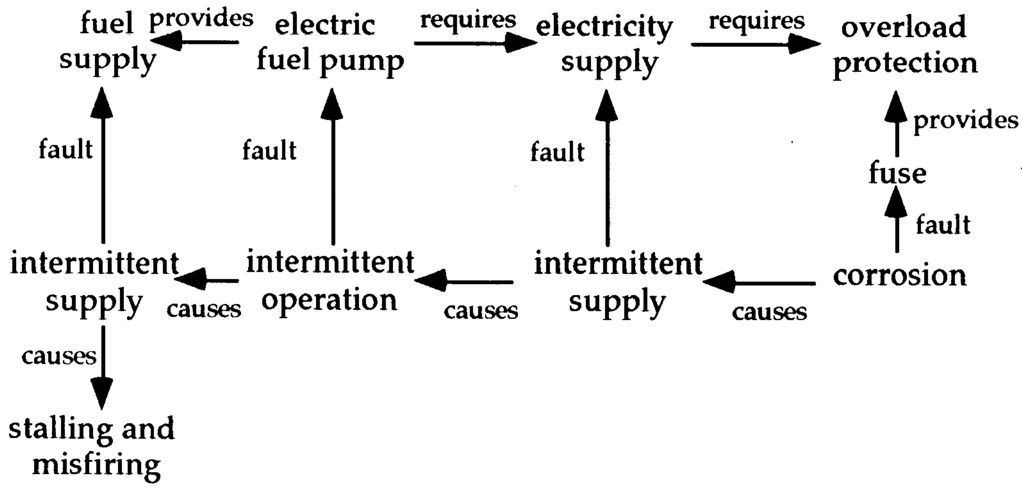

Figure 4.3. Explanation of fault in "corroded fuse" story.

is analogous to the student. This person is chosen through an examination of the roles given to the individuals mentioned in the index. The student in the simulation is playing a particular role: car mechanic, salesperson, etc. The person in the story who is playing a role similar to the student's is an important figure for determining the kind of analogy the telling of a story will create. I call this person the *student analog*.

4.3.2. Creating the trigger

The trigger is built from the social and physical context of the story as represented in the index. However, it is certainly too restrictive to demand that a story only be told when the student's context matches its physical and social setting exactly. In the case of the "Corroded fuse" story, it would not make sense to tell the story only when the student is working on a Ford pickup because this problem could occur in many other cars. The problem could not occur if the car were one where the fuel pump does not have a fuse: if the pump were powered mechanically, for example.

So SPIEL must generalize the social and physical setting of a story. The reason that the type of fuel pump is important and the model of car is not has to do with the causal relationship between these aspects of the setting and the events that are central to the story. See the explanation in figure 4.3. The fuel pump requires a supply of electricity, which passes through the fuse. Corrosion on the fuse holder causes an intermittent supply of electricity, resulting in an intermittent supply of fuel. This causes the symptoms of stalling and misfiring. The fact that the fuel pump is electric is obviously essential to this explanation.

The identification of important aspects of features by finding their role within an explanation in explanation-based generalization (Carbonell, 1983). For SPIEL, this type of generalization is made somewhat easier because the index already contains explanations. It may have to elaborate slightly to see whether a feature of the setting is important to the point of the story that the index describes, but it does not have to construct a new explanation. Generalization is performed in SPIEL by dropping features of the setting that do not have a role in the explanations in the index. It could drop "manufacturer of car" as a feature of the setting, but keep "electric fuel pump."

It might be possible to do even better, to reason about how far to generalize those aspects of the setting that are found to be important. A storyteller with a lot of knowledge about cars and corrosion might note that the reason for the corrosion was excess moisture remaining on the fuses, and that cars whose fuse boxes are located in the passenger cabin would not be subject to this problem. An even better generalization than "cars with electric fuel pumps" might be "cars with electric fuel pumps and external fuse boxes."

The amount of generalization of storytelling conditions that a system can perform is a function of how much it can explain about the events in its stories. Since SPIEL's indices are already couched as explanations, it gets some explanation for free. Additional elaboration on top of what is contained in the index is quite difficult, however. Causal explanations such as that found in figure 4.3 require a causal theory for their construction. Explanations of phenomena in social interaction would require a broad causal theory of social cognition and interaction, the development of which is a long-term research project in itself. SPIEL's storytelling strategy perform small bits of explanation where they must, but the system has no general explanation capacity.

4.3.3. Reasoning about evidence

The evidence part of the RCD is what describes the actual conditions of relevance. Creating the evidence conditions requires reasoning about evidence gathering, determining what should count as evidence for or against the recognition of the tutorial opportunity. Every storytelling strategy has its own method for reasoning from an index to a description of the evidence needed to recognize a storytelling opportunity involving the strategy and the story represented by the index. The "Provide alternate hypothesis" strategy used in the "corroded fuse" example calls on the storyteller to tell a story about a fault that appeared to be caused by part X, but was actually caused by part Y, when the student has experienced similar symptoms, and diagnosed part X as the source.

Some of SPIEL's storytelling strategies involve a more complex relationship between story and RCD. The **Demonstrate risks** strategy calls on the storyteller to tell a story about a failure when the student has experienced success. The evidence conditions produced by this strategy look for the student to encounter a result opposite to the one represented in the index. To create such conditions, the strategy must be able to reason about opposites.

All storytelling strategies have to reason about how concepts in the index, or those inferred from the index may be manifested in observable actions in the simulation. In some cases, what is in the index is easily observed, such as whether or not a student makes a sale. In other cases, more inference is required. For example, in order to recognize that a student car mechanic has diagnosed a fuel pump as faulty, SPIEL must look for the student to manifest this decision by attempting to replace the part or taking some other observable action.

4.3.4. Generalizing evidence

In the evidence conditions, the same question of generalization arises as it did for the trigger. How specifically must a student's situation match the story? Suppose the student were working on a car with a headlight that worked intermittently. If the student chooses to replace the light, should a storyteller bring up the "corroded fuse" story?

On one level, this would make a great deal of sense. Both the headlight and the fuel pump are electric parts using fuses that might corrode and result in intermittent operation. However in most cars, there is a single fuse for both headlights. If the fuse really were at fault, they both would be affected. The fix suggested by the story is not compatible with a single faulty headlight, and it would be wrong to tell it in such a situation, but in order to know this, a storyteller would have to have a great deal of knowledge about typical designs of automotive electrical systems.

Similar issues arise for stories about an area of social expertise like selling. Suppose SPIEL has a story about getting the customer to buy an increased advertising program. If the student is trying to sell a renewal, should the story be considered similar or not?

To a layperson, getting a renewal for an ad looks a lot like getting the customer to increase advertising. However, just as an intermittently-functioning headlight superficially resembles a problematic fuel pump, an analysis of cause and effect shows that these surface similarities hide a deeper difference. Arguing for a renewal is a different persuasion problem than arguing for more advertising. Different objections will arise and different counter-arguments will be called for. Again, knowledge of the domain must enter into decisions to generalize.

Although the problem is the same as for the trigger condition, the solution is not. Evidence conditions cannot be generalized through reference to the explanations already existing in the index. The elements of the index involved in the evidence conditions, such as the goal, plan, and result are already part of the explanations in the index. SPIEL will not be able to rule out any of them. The only other alternative is to construct more general versions of the evidence conditions while preserving the causal structure of the index. This would require additional explanation elaborating on the explanation in the index. A system with a good model of how cars operate and how they break could perform such reasoning in the domain of car repair, but as I have discussed, such a model is not available for the areas that SPIEL teaches. In absence of this kind of causal knowledge, SPIEL cannot generalize the recognition conditions that gather evidence for the tutorial opportunity. It stays close to the level of specificity found in the index.

4.3.5. Visible vs. hidden evidence

Another important issue in the generation of evidence conditions is the disparity of knowledge between the student and the tutor. A tutor in a case-based teaching system can inspect areas of the simulated world that are hidden from the student. If SPIEL infers that a student is hoping that a character in the GuSS simulation has certain intentions, it has the capacity to inspect the simulation data structures and find out if, in fact, that character has the desired intention or not. In fact, SPIEL cannot avoid using this capacity, since when characters make utterances, the student sees an English language string, which SPIEL does not have the capacity to understand. SPIEL can only look at the internal, hidden, representation of the utterance.

So, SPIEL does inspect the internals of the simulation; the question is how much it should use this capacity. There are two dangers here. If the storyteller intervenes based only on the external appearance of the simulation, it may miss some obvious hidden cue that would tell it the story is not appropriate. In our car repair example, the fuel pump really might be faulty. The storyteller would then appear to give bad advice, discouraging the student from making the correct repair. However, if the student gets only those stories that really apply in his or her current situation, the storyteller may end up giving away clues about the problem. If the student knows that the storyteller only intervenes when it has verified its analogy through reference to the hidden structure of the simulated scenario, then the storyteller may inadvertently give away information about the scenario that the student would otherwise have to work to uncover.

This is a difficult dilemma in which either alternative presents some problems: use only visible evidence and risk giving bad advice, or use hidden evidence and risk creating an unrealistic practice environment. SPIEL's solution is to use visible evidence as much as possible, relying on hidden evidence only when the risk of bad advice is high. In most cases, the risk of bad advice is not high. SPIEL therefore does not look for hidden evidence to verify conditions that are suggested by external manifestations: even if the fuel pump is at fault, the "corroded fuse" story gets told.

This is not as hazardous as it may seem. Unlike a tutor that is giving directive advice, SPIEL is making analogies between students' situations and its stories. Students always have to interpret what the storyteller says. They must compare the actions described in the story against their own actions and determine if the story is a good analogy. A student who has carefully tested the voltage input to the fuel pump would be well satisfied that the fuse was not at fault and not change course as a result of the story. A student who had not checked the voltage would perform this test, find everything normal, and continue replacing the pump.

Sometimes, however, the danger of bad advice is high. Several of SPIEL's storytelling strategies concern themselves with the viewpoint of others in the simulation, unlike the majority that seek to explain the circumstances of a person in a role similar to the student's. In strategies which explain the situations of other characters, students cannot easily determine the quality of a story's intended analogy. This makes reliance on the external manifestations of internal states risky. Because the student cannot check the analogy, bad advice is more likely to be taken to heart. Therefore it is important that SPIEL verify that the characters in the simulation are motivated in the way that a story suggests. It looks directly at the characters' internal states to gather evidence for this. This may result in some unwarranted hints to the student, but the danger of bad advice makes this risk worth taking.

Hidden evidence is also used to resolve conflicts between storytelling strategies. Suppose a student prepares to convince a customer about the benefits of geographic expansion, and hears a story reinforcing her approach, a story about someone who convinced a similar customer to buy ads in order to expand. The student puts this plan into action, but finds that the customer is not receptive and the sale is not successful. True independence between storytelling strategies would dictate that the same story should be told to show an opportunity that was missed: "You didn't succeed, but here's a story about someone who did..." This would probably be quite annoying. Since the student has already seen the story once, its re-presentation merely emphasizes the failure.

Pragmatic conversational considerations therefore require that the strategies that project possible outcomes try to avoid conflict with those that show alternative outcomes. This conflict can be detected if there is hidden evidence of a plan's failure before that failure has become evident to the student. In the geographic expansion example, the system might be able to detect that the customer is not actually interested in geographic expansion. It would then hold off telling the story until the failure became apparent.

4.4. Rule generation

Creation of the trigger and evidence conditions completes the construction of an RCD. The RCD represents the recognition conditions for a tutorial opportunity describing a good time to tell a particular story using a particular strategy. The next part of the storage-time processing of SPIEL is rule generation, the construction of GuSS decision rules capable of recognizing these tutorial opportunities.

For an example, consider an RCD generated for the story "Long-term goals." This is a story about a salesperson who probes the customer for information about his long-term goals in business and then uses this knowledge to construct an effective sales presentation. The **Demonstrate alternative plan** strategy uses this story to demonstrate this tactic when the student has failed to gather any information about a customer's long-term goals. The RCD looks like this:

WHEN the student is in the pre-call stage, and
speaking to someone who is the decision maker,
LOOK FOR the student to perform cursory information gathering
about the customer's long-term goals, and
the student to fail to find out the customer's long-term goals.
THEN TELL "Long-term goal"
AS a "Demonstrate alternative plan" story.

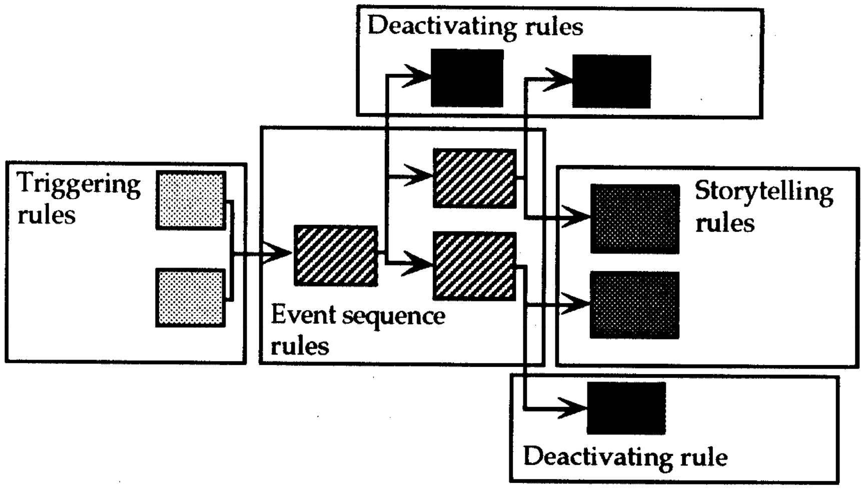

Figure 4.4. Schematic of a rule specification

4.4.1. The expanded RCD

The first step in rule generation is to make the recognition conditions concrete, to reason about how these conditions might arise and describe their occurrence in terms of actions that the student might actually take. Sometimes a recognition condition is something that takes place over time. The example RCD contains the action of the student performing cursory information gathering about the customer's long-term goals. Several steps must take place before such an action can be recognized: the student must be in a conversation with the customer, and that conversation must end without the student having gathered much important information about long-term goals. Other actions may have more than one possible realization, for example, the end of a conversation can occur when one of the participants leaves the room or when the conversation is interrupted by someone else.

To perform this elaboration, SPIEL needs knowledge about observability in the simulated world. It must know what kinds of actions are and are not available to the student and the simulated agents. For abstractions that cannot be immediately recognized, it must be able to construct sets of observable features that would constitute that abstraction, if recognized in the appropriate context. The output of this step is an expanded RCD (or eRCD) representing the same recognition conditions as the original RCD, but placing them in terms of their concrete, observable manifestations. Here is the eRCD for the RCD above:

WHEN the student is in the pre-call stage, and
speaking to someone who is the decision maker,
LOOK FOR the student to leave without asking about the customer's
long-term goals, or
the student to ask about long-term goals, but not find out
anything about them, and then leave.
THEN TELL "Long-term goal"
AS a "Demonstrate alternative plan" story.

Expansion has taken place in the evidence conditions. The test for cursory information gathering has been made more concrete, turned into two tests, each of which looks for a sequence of actions on the part of a student.

4.4.2. The rule specification

From the expanded RCD, the system designs a set of rules that will recognize the situation it describes. The initial stage of rule design is the construction of a *rule specification*, a directed

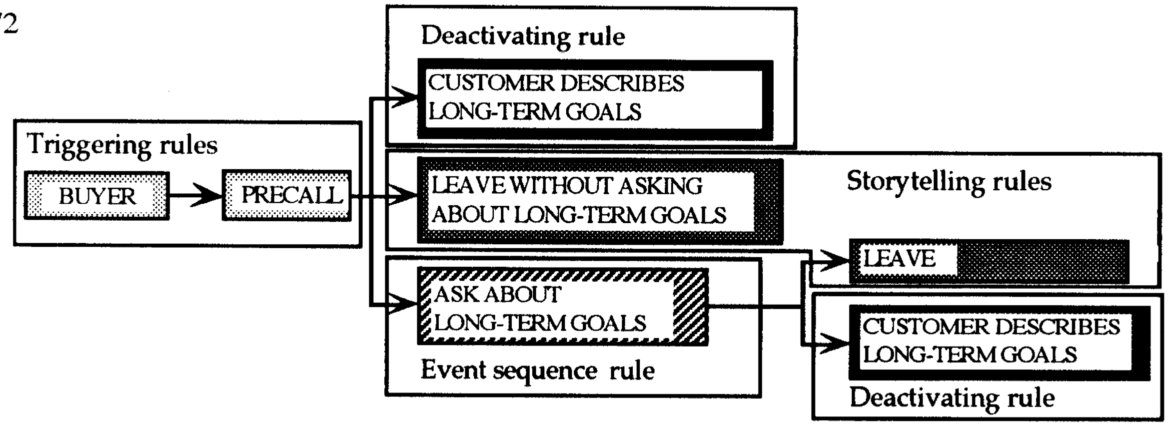

Figure 4.5. Rule specification diagram for "Long-term goals" example

graph indicating what production rules are needed and how they will relate to each other. An example is shown in figure 4.4. Each node in the graph is a rule that the generator will have to produce. The connections in the graph represent rule sequencing: if rule A enables rule B, then there is a connection from A to B in the rule specification. Multiple outgoing paths indicate exclusive disjunction: if there is a connection from A to several nodes, B, C and D, then only one of rules can actually fire after A, whichever one's conditions are met first. Multiple incoming paths represent conjunction: if there is a connection from A and B to C, then both rule A and rule B must fire before C will be enabled.

There are several different types of nodes in a rule specification as the figure shows. The triggering rules recognize the context for storytelling. They are created from the trigger parts of the RCD. The event sequence rules, corresponding to the evidence conditions, form a branching sequence of rules that fire in sequential order, each looking for the next temporally-ordered step in the sequence of conditions leading to recognition. The last step in the recognition sequence is a storytelling rule that causes the simulation to retrieve the story and signal the student that it is available.

Deactivating rules hang from this recognition sequence. If the context established by the triggering rules is violated in some way, the opportunity cannot be recognized and the recognition sequence must be aborted. For example, if an RCD depends on the fact that a salesperson is making a presentation to someone other than the decision-maker, then the arrival of the decision-maker will destroy the "absence of decision-maker" context that makes the story relevant. At this point, SPIEL must stop looking for an opportunity to tell the story until the appropriate context reappears.

The initial creation of the rule specification is a straight-forward transformation from the expanded RCD. However, once this initial rule specification is generated, two knowledge-based simplifications are performed. (1) SPIEL eliminates conditions that can be assumed given the nature of the student's role. For example, in some contexts, the fact that the student is trying to make a sale can simply be assumed. (2) SPIEL merges tests that can be performed in a single step, using some knowledge of how conditions become true in the simulation. If it needs to recognize the student's entrance at a location and the presence of a certain person at that location, it can merge these into one rule that, when the student enters a location, tests for the presence of the person.

A diagram of the rule specification for the "Long-term goals" example is shown in figure 4.5. The triggering conditions are implemented by two rules, one that looks for the individual who is the buyer, and one that looks for the student to begin the "pre-call" (information-gathering) step of the sales process. If the customer talks about his or her long-term goals in the course of

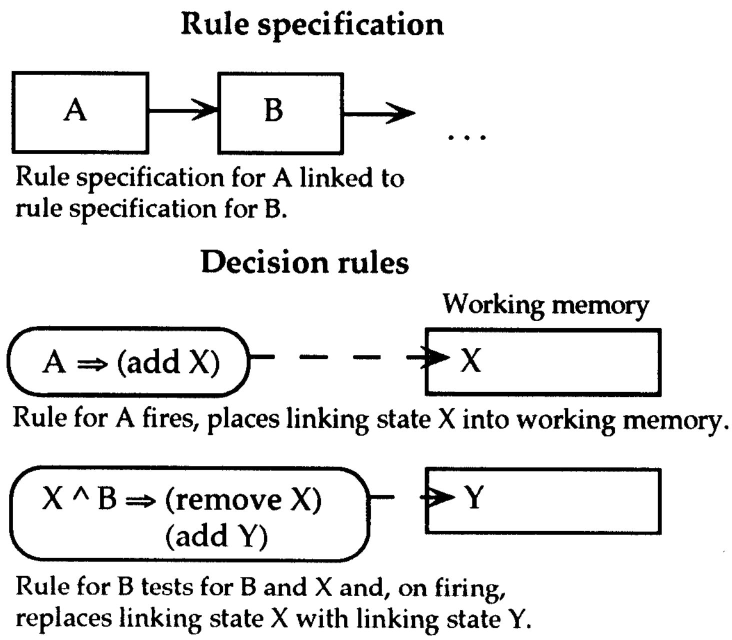

Figure 4.6. Implementing connectivity in GuSS

the pre-call conversation, the story is not relevant, because the student has succeeded rather than failed at finding out about the customer's long-term goals. There is a deactivating rule that halts the recognition attempt if this happens. If the student leaves without asking the customer about long-term goals, this certainly counts as cursory information gathering on this topic, so the story gets told. A third possibility is that the student does ask about the customer's long-term goals. If there is a direct response, then the story will not be told, but if the student never gets a straight answer, it will.

4.4.3. Rule creation

In the final rule specification graph, each node contains a condition or set of conditions that a rule will have to recognize. Rules are produced by walking through this graph translating from the descriptions into simulation states that rules can directly test for. For example, a "student entering a location" condition becomes a test for an entering event with the actor and destination slots bound to certain values. Chapter 9 describes SPIEL's translation of its MOP-based representations into the vocabulary of GuSS simulation states

Recall that the teaching modules in GuSS, like the simulated social agents, are implemented by production systems. The enabling relationship between rules, represented by the connectivity of the graph, is implemented through the working memory of SPIEL's production system. See figure 4.6. Each rule is linked to following rules by the placement of a *linking state* into working memory. Each linking state is unique and has no other function than to ensure proper rule sequencing. Connected rules, regardless of what other conditions they test for, must check for the presence of the linking state before being allowed to fire. This preserves the rule sequence. They must also remove the linking state upon firing to preserve the property of exclusive disjunction.

4.4.4. Optimization

Once the stories in the library have been through the processes of RCD creation and rule generation, SPIEL has a collection of *rule sets*. Each rule set is a collection of production rules, designed to recognize a tutorial opportunity – a chance to tell a story using some strategy. The elaborate-and-index model used to derive these recognition rules treats each

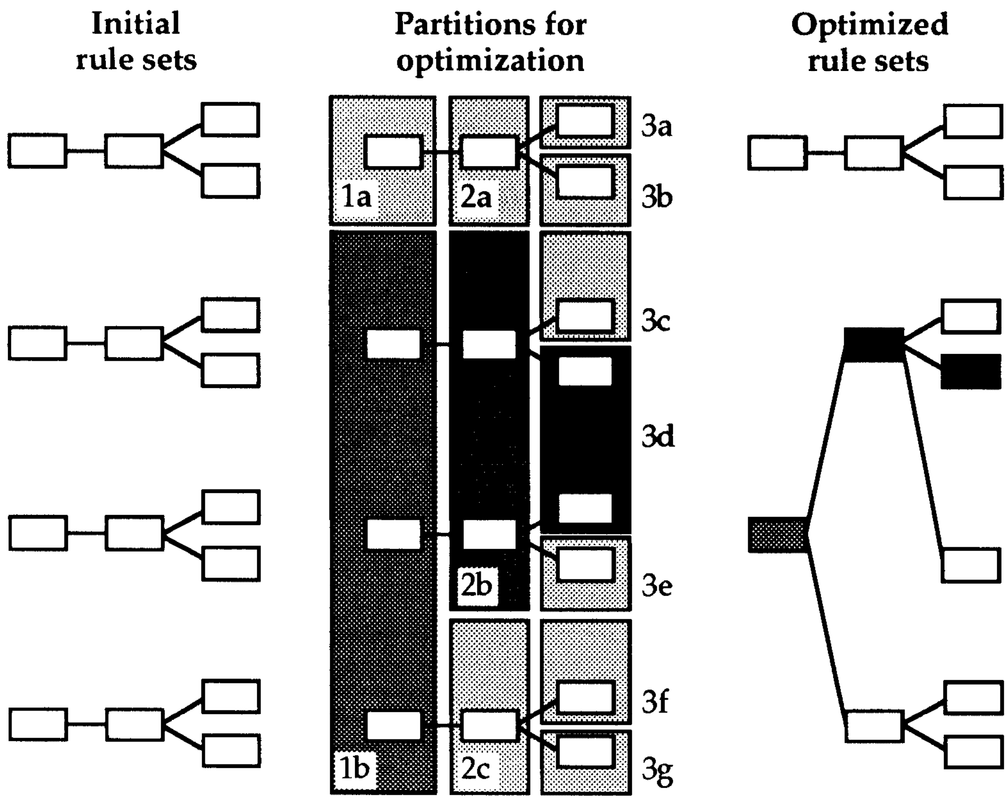

Figure 4.7. Optimization example

index separately and the rule sets are therefore autonomous. They do not communicate with each other. This means that there is considerable redundancy between rule sets. There may be several stories that are relevant when a student is selling to a disgruntled customer, but the rule sets that look for opportunities to tell these stories would each have their own tests for disgruntled-ness.

To eliminate this duplication of effort, SPIEL optimizes, trying to merge rules that perform the same tests into a single rule. Starting with the rules in each rule set that have no predecessors, it partitions the rules into equivalence classes containing rules with identical left-hand sides. When there is more than one rule in a partition, it creates a new rule that subsumes the entire partition. The process is then repeated for the rules that follow in sequence. In this way, independent rule sets can be made to share rules where possible.

Figure 4.7 shows a simple optimization example. For simplicity, the four initial rule sets, shown at the left, all have the same structure. The partitioning of the rule sets is shown in the center. Of the four rule sets, three of them have identical initial rules and are grouped in partition 1b. These can be folded into a single rule, shown by the same-color node in the final optimized version of the rules shown at the right. There are three successors to this new rule and of these, two are identical, as seen in partition 2b. They are combined to create the next merged rule. The next set of successors is then examined, yielding another case of overlap in partition 3d, which produces the last optimized rule. In the final rule collection shown at the right, the top rule set is unchanged, since none of its rules had anything in common with the others. The other rule sets have been optimized: out of each partition containing more than one rule, only a single rule remains. The final collection has 12 rules where the initial one had 16, a savings of 25%.

4.5. Natural language generation

Storytelling strategies determine when a story should be told. They also dictate how it should be told. Recall that each of the examples of tutorial storytelling contains three pieces of text that explain the system's reminding to the student:

Headline: The headline is the first notice to the student that a story is available. It gives just enough information so that the student can tell what kind of story is coming. For example, "A story about a different approach you might try."

Bridge: The bridge describes the connection that SPIEL seeks to draw between the story the student is about to see and the situation the student was just in: "You gathered information about long-term goals in a cursory manner without much success. Here is a story about a similar situation in which an account executive used a different method and was successful:"

Coda: The coda helps the student apply the message of the story to the situation that caused it to arise: "You gathered information about long-term goals in a cursory manner. It didn't work well. In the future, you might consider probing deeply into the customer's long-term goals."

Labov (Labov 1972) uses a six-part structure to describe conversational narratives: abstract, orientation, complicating action, evaluation, result, and coda. The first and last of these are most important for fitting the story into the context of its presentation. I use the word "bridge" rather than "abstract" to emphasize the function of the story's introduction as helping the student make the transition from interacting with the simulation to viewing the story. I have kept the term "coda" to refer to the final recapitulation. The headline is an artifact of GuSS's interface design and does not have an exact counterpart among descriptive theories of conversation.

These texts can be generated in a standardized way using templates associated with each storytelling strategy. For **Demonstrate alternative plan,** the three templates are:

Headline: "A story about a different approach you might try."

Bridge: "You <plan> without much success. Here is a story about a similar situation in which <gen-agent> used a different method and was successful:"

Coda: "You <plan>. It didn't work well. In the future, you might consider <story-plan>."

Items in angle brackets, such as <plan>, are used to instantiate a template for a particular story. The headline is often sufficiently stereotyped that it does not need to be customized for each story. An exception is the headline for **Explain other's plan**. It reads "Do you understand what <actor> is doing?" Filling in this template lets the student know whose actions the story will attempt to explain.

The same mechanisms that SPIEL uses to determine what tutorial opportunity it should try to recognize also can be used to instantiate the templates. For example, when applying the **Demonstrate alternative plan** strategy, SPIEL identifies the plan that it will look for the student to have, "cursory information gathering." It can translate that into an English language phrase using a table of English equivalents for its MOPs and fill in the template.

In the case of the **Demonstrate alternative plan** strategy, all of the fillers in the template can be computed at storage time.

<plan> is the "opposite" plan that the system seeks to recognize the student using.

<story-plan> is the successful plan that the story describes, probing the customer's long-term goals.

<gen-agent> is a generalization of the agent in the story. Mr. Beam is the agent in the index, but he is generalized to become "account executive."

Other fillers must be instantiated at the time the story is retrieved. In the "Wife watching TV" story, the character that the student appears to have the assumption about can only be identified at retrieval time. The templates, in whatever stage of instantiation, are passed along through the rule generation process, first as part of the recognition condition description, until they are placed within the telling rules.

Where I show "TELL 'Long-term goals' AS A 'Demonstrate alternative plan' story," for simplicity, there is actually a more involved TELL clause, that incorporates the explanatory texts. A more complete description of the telling action in the RCD for the "Long-term goals" story would be as follows:

TELL "Long-term goals" USING
HEADLINE: "A story about a different approach you might try."
BRIDGE: "You gathered information about long-term goals in a cursory manner without much success. Here is a story about a similar situation in which an account executive used a different method and was successful:"
CODA: "You gathered information about long-term goals in a cursory manner. It didn't work well. In the future, you might consider probing deeply into the customer's long-term goals."

A story that addresses the student's expectations, such as "Wife watching TV," is a somewhat more difficult case for the natural language generator. What is important in telling this story is that the system has recognized that the student may have an assumption similar to the anomaly at work in the story. The bridge has to explain both parts of an anomaly. For anomalies, the generator must use some of the same reasoning that is applied in the **Warn about assumption** strategy. (See Chapter 6.) There are generation links associated with certain MOPs that highlight parts as important for description. In "theme," for example, the relationship to business decision-making is considered important for describing a theme. Also important is whether a theme is one that is associated with an individual, like a personality trait, or one that is associated with a role, a role-theme.

In the case of "Wife watching TV", the important difference between themes can be summed up by the difference in business role. The housewife role is not associated with business decision-making and the business-partner role is. So, the translation for the assumption is "will not have a role in the business."

4.6. Related work in retrieval

4.6.1. Cue-based retrieval

SPIEL's retrieval occurs through a process of opportunity recognition. Storage-time processing defines a set of storytelling opportunities and retrieval-time processing attempts to recognize them. This is quite different from what is usually thought of as retrieval in computer science. The term "retrieval" usually means *cue-based retrieval.* Cue-based retrieval involves two processes: cue formulation and search. During the cue formulation step, a situation is analyzed to compose a cue, a general description of the kind of information that is being sought by the retriever. The search process takes the cue and attempts to find the item or items that best match it.

The search process has received a great deal of attention in information retrieval research, which has concentrated particularly on the problem of making fast matches in huge libraries of simple indices (Salton and McGill 1983). Information retrieval systems, for the most part, leave the creation of the retrieval cue up to the human user.

Case-based reasoning systems do perform their own cue formulation. Many different architectures have been proposed (see (Kolodner 1993) for a survey), but in general, CBR systems use cue-base retrieval, using a body of heuristics for analyzing problem situations and deriving retrieval cues. For example, in CHEF, a case-based planner for cooking (Hammond 1986), the system extracts certain aspects of the desired dish to be produced, such as the desired taste, texture, and ingredients, and uses this cue to search a conceptual hierarchy for recipes that produce similar dishes.

The problem with strict cue-based retrieval is its dependence on heuristics for analyzing situations. A cue-based retriever will not be able to find a good case to retrieve unless it can understand the situation well enough to know what features are important and therefore worth using as a retrieval cue. As discussed in chapter 2, SPIEL's domain of social interaction is one where there are few hard-and-fast rules that can be relied on when analyzing social situations.

SPIEL is also in a slightly different position from a cue-based retriever in that it is not serving the needs of some other process. The classical IR model depends on user initiative: the user must recognize a knowledge need and seek to address it. Case-based retrieval models generally depend on the initiative of the problem-solver, which requests a case while working on a new problem. In SPIEL, it is the stories themselves that determine how and when the tutor should take initiative. This makes opportunity recognition an attractive retrieval model.

4.6.2. Alternatives to cue-based retrieval in case-based reasoning

Some case retrievers do have alternatives to the cue-based model of retrieval. Owens' (1990) case retriever ANON has what he calls an integrated memory. It retrieves explanations of plan failures in response to planning problems, but rather than formulating retrieval cues from problem descriptions, ANON uses its cases to determine what features have the most discriminatory power in a given situation. That is, it looks for the feature that, if recognized as present or absent, would divide the set of cases under consideration most nearly in half. It then attempts to recognize that feature and reduce the number of cases it is looking at. This "divide and conquer" retrieval strategy is incremental, in that features are successively derived from the problem situation and the set of relevant cases successively narrowed.

SPIEL's need for incremental retrieval is subtly different from ANON's however. SPIEL must combine many different fragments of retrieval information whose occurrence is distributed over time. SPIEL's incremental retrieval is governed not by the needs of the retriever, but by the environment, which dictates when features will be available to be recognized. This kind of temporally-incremental retrieval is not often found in retrieval systems, although Riesbeck (1993) proposed a "shoot first, ask questions later" approach to case retrieval that does have this property. Retrieval strategies, such as SPIEL's storytelling strategies, are crucial in such a system, because they give the retriever the ability to identify when and where relevant features might appear.

4.6.3. Opportunistic memory

Retrieval as opportunity recognition is very similar in spirit to the notion of opportunistic memory, put forward by Hammond in the RUNNER system (Hammond 1989). The RUNNER system is an opportunistic planner that prepares itself to act using something akin to the elaborate-and-index model of opportunism. When RUNNER realizes that it has a goal that cannot be achieved in the immediate environment, it devotes some time to deciding what an opportunity would look like to work towards that goal. These recognition conditions continue to be monitored as RUNNER pursues other goals, so the suspended goal can be recalled when an opportunity to achieve it arises. This is similar to SPIEL's opportunistic tutoring behavior.

SPIEL's retrieval mechanism also has similarities to Minsky's agent-based model of intelligence (Minsky 1986).[1] Each story can be considered an agent that actively seeks its own telling. SPIEL's storage-time process of generating recognition conditions is an example of what Minsky sees as a process of delegation. To get themselves told, stories must rely on sub-agents that have the ability to identify storytelling opportunities in the learning environment.

4.6.4. Retrieval strategies

Another important aspect of SPIEL's retrieval that is different from cue-based retrieval is the need for a variety of retrieval strategies that characterize different kinds of relevance. The cue-based model puts all of the system's reasoning about relevance into the cue formulation step. However, as chapter 2 discusses, for SPIEL, this would entail building an "expert model" of social reasoning capable of diagnosing the student's failings, in order to formulate cues related to tutorial goals. A cue-based solution would fail to take advantage of SPIEL's story base as a case-based model of expertise: that knowledge would have be duplicated in the cue formulation process.

Because most case retrieval systems put their relevance decisions into the cue formulation step, they can employ a simple similarity metric to evaluate the fitness of cases (Bareiss and King 1989). In general, the problem of comparison in retrieval always amounts to a utility metric that evaluates a case's applicability in a certain situation (Kolodner 1989). It just so happens that for most problem-solving situations, similarity assessment is a good approximation of a utility metric. This is not the case in tutoring.

Each storytelling strategy in SPIEL defines a different utility metric, a structured comparison in which similarity may be important in some places, but not in others, and in which explicit dissimilarity or other relations are used. Such utility metrics will become increasingly important as case-based reasoning comes to be applied to a wider class of problems, for example, the domain of design (Kolodner and Wills 1993; Wolverton and Hayes-Roth 1993).

4.6.5. HYPO

One of the first case-based reasoners with explicit retrieval strategies was HYPO (Ashley 1990). HYPO built legal arguments by reasoning about legal cases. It retrieved cases to support its own arguments, but also looked for good opposing cases so that it could prepare itself to counter arguments likely to be presented by the other side. It had essentially two case-usage strategies: find a case that supports a certain position on a certain issue, and find a case that argues against it. To find cases for these purposes, HYPO formulates a cue based on the legal properties of the case under consideration and performs retrieval based on similarity along certain dimensions. It then builds a structure, the claim lattice, of the retrieved cases, that enables it to identify contrasts along other dimensions and find cases that support or oppose its position. This approach works for legal reasoning, because most pieces of the retrieval cue are the same for supporting and contradicting legal cases. For the wider variety of storytelling strategies used by SPIEL, the initial step of similarity-based retrieval makes less sense since there are few similarities that are common among all of the strategies.

Ashley and Aleven (1991) discussed an extension to HYPO in which the program is used to tutor students in the development of legal arguments, by critiquing their choices of cases for various roles in an argument. This version of HYPO uses a library of explicit case employment strategies, ranging from "relevantly similar" to "best case to cite." Although all of these strategies rely solely on different kinds of similarity measures, this tutorial version of HYPO is an example of the importance of explicit representation of comparison strategies in retrieval, particularly for education, where a tutor must explain what it retrieves.

---

[1] The author took this perspective explicitly in earlier research on conversational story retrieval (Burke 1989).

4.7. Storytelling strategies

Storytelling strategies govern the elaboration of tutorial opportunities in SPIEL's elaborate-and-index model of opportunistic retrieval. They dictate how stories will be presented and at what times. In this capacity, they determine the conditions under which stories will be considered relevant to the student's situation. Each strategy aims for a different kind of relevance, drawing its own kind of analogy between story and storytelling situation.

In order to reason about the relevance of stories, strategies need the ability to compare. SPIEL's storytelling strategies are, in part, a function of the types of comparisons that can be drawn between its representation of a story (the index) and its representation of the student's situation (the GuSS simulation). The next four chapters describe SPIEL's 13 storytelling strategies, grouping them in ways that emphasize the different kinds of assessments of similarity and difference that they make.

Similarity of action, difference in outcome (Chapter 5): Since SPIEL can compare what the student is doing against what others in its stories have done, and since it can contrast outcomes, it has a set of strategies that show the possibility of an alternative outcome: **Demonstrate risks**, **Demonstrate opportunities**, and **Demonstrate alternative plan**.

Similarity of expectation (Chapter 6): Six other strategies compare the student's expectations against the expectations found in stories in order to critique the student's perspective. There are two forms of each of the strategies **Warn about optimism**, **Warn about pessimism**, and **Warn about assumption**.

Similarity of action, outcome unknown (Chapter 7): This chapter returns to the theme of finding actions similar to those found in stories, describing two strategies that project the possible results of student actions: **Reinforce plan** and **Warn about plan**.

Similarity of other's intention (Chapter 8): The final two strategies compare the situations of characters other than the student with situations described in stories. These strategies help the student understand the perspectives of others in the simulated world. They are **Explain other's plan** and **Explain other's perspective**.

5. Stories That Show Alternatives

5.1. Alternative-finding strategies

> Betty: "I'm having so much trouble with the Fields account. Nobody there will return my calls."
>
> Dolores: "Really. When I worked that account last year, I told the secretary that the advertising in the legal services heading in the Metro directory had really changed over the past year, and that their program needed to be updated. Mr. Fields called me the next day."

This conversation shows how a story can be used to illustrate an alternative. Betty describes her situation. In response, Dolores tells a story about a similar situation in which the outcome was much different: a successful attempt to get through to a customer.

Whenever one confronts a complex problem, it is important to know the range of possible outcomes: What is the best and worst that could happen? What is reasonable to expect? Both in real life and in GuSS's simulator, the student gets to see only one outcome at a time. Stories that show a range of alternatives provide necessary balance, and delineate what might be expected, based on other people's experiences. The alternative-finding storytelling strategies discussed in this chapter are designed to bring stories to the student's attention when those stories differ in significant ways from the student's experience in the simulated world.

SPIEL uses three strategies to find stories that show alternatives. **Demonstrate risks** tells stories about failure when the student is in a similar situation, but has experienced success. **Demonstrate opportunities** acts, like Dolores in the example above, to give positive examples of success when the student has encountered failure. The third strategy that uses alternatives, **Demonstrate alternative plan**, looks for an alternative to the plan that the student has followed, to show the student a new way to do something.

As I describe these strategies, I will discuss their general educational purpose and give an example from YELLO that shows how the strategy operates. I will then describe the strategy's storage-time operation: what stories it selects and what operations it performs on indices to create the trigger and evidence portions of the RCD.

Of particular importance for the strategies in this chapter is the mechanism for alternative-finding inference. What counts as an alternative is a function of the situation in which the alternative is sought. Alternatives cannot always be determined by a simple set of links between concepts and their opposites. I will describe various aspects of the opposite-finding inference algorithm as the functioning of each strategy is discussed.

5.2. Demonstrate risks

> Dan: Dr. Hartfield just bought a half-page ad. He really was sold by that testimonial letter from Dr. Graves.
>
> Bert: That's amazing. Last time I showed that letter to a doctor, he got really upset. He said that Graves was a black spot on the medical profession. I almost lost all his advertising because he didn't want to be associated with that kind of doctor.

The idea behind **Demonstrate risks** is to tell a story about a failure when the student has experienced success. One reason this is important is that no simulation is perfect. If the simulation is not accurately reproducing the circumstances encountered in the story, the story points out possible differences between the simulation and the real world of practice. Another way that a **Demonstrate risks** story may arise is if the student successfully uses a risky strategy, one that fails nine out of ten times. By showing how a situation similar to the one

- The student has successfully sold an ad and got the client's signature.
- The student keeps talking to the client for a little while before leaving.

A story showing the risks of your approach.

You kept talking to the client after the sale was closed. Nothing bad happened but here's a story in which doing that led to problems:

I was in the South Bend/Mishawaka area. This was my first or second year. I was calling on a swimming pool contractor. He had quarter page in South Bend. I was proposing a quarter page in another directory. It was sitting at his kitchen table. And the guy was hesitating; he didn't know... So, after a few more attempts, he says to me "OK, I'll go with the other quarter page." He bought it. I pushed the order form in front of him. He signed it. It's done.

As I'm putting my stuff together, I made this comment that cost me the quarter page. I said, as I'm packing up, "I'm sure that you're going to get a lot of phone calls from that new ad." He looked at me and he said, "You know, I don't need any more phone calls. I'll tell you what, let's not do that this year, maybe next." I talked myself out of a quarter page. I've never done it since. I walked out. There was nothing I could say. I had it and I lost it. All I had to say was "Thank you very much Joe. See you next year." But I didn't. I had to tell him about the calls, which I'd already done twenty times.

Nothing bad happened to you because you kept talking to the client after the sale was closed, but sometimes the client changes his mind.

Example 5.1. Telling "Talked myself out of a sale" using the **Demonstrate risks** strategy.[1]

encountered in the simulation could have been a failure instead of a success, the storyteller demonstrates that a successful result in the simulation is not always repeatable in the real world.

Example 5.1 shows **Demonstrate risks** in action in YELLO. Here SPIEL observes the student continuing to converse with the client after closing a sale. This is a risky move: every additional minute the salesperson hangs around is an opportunity for the customer to have a change of heart. When SPIEL tells a story about a customer who backed out of a deal in a similar situation, the student gets a chance to learn about this risk.

5.2.1. Applying the strategy

"Talked myself out of a sale" is a good story to demonstrate risks to a student because it shows a negative result of something a salesperson did, and can be used if the student does the same thing but achieves a positive result. More generally, a story is a possible candidate for telling using the **Demonstrate risks** strategy if the Actual part of the index contains a negative result and the agent in the index is a student analog.

Filter for Demonstrate risks: Select stories whose indices show the student analog achieving a negative result.

Figure 5.1 shows the index for "Talked myself out of a sale." The anomaly in this index has to do with outcomes: the salesperson wanted to reassure the client and gain his confidence, but he ended up losing the sale. This difference in outcome is the most salient difference between the salesperson's hopes and the actual events of the story. There are different goals at work, as the explanations in the columns indicate, but the story is not about an anomalous goal. It is not the case that the salesperson wanted to have the goal to obtain the client's confidence, but ended up

[1]Story from interview with Denny Gant, video clip #150.

Physical setting:	close, service, small
Social setting:	sales target
Viewer:	seller
Perspective:	wanted
Agent:	seller
Anomaly Type:	result

	Wanted	*Actual*
Theme:	seller	seller
Goal:	obtain client confidence	make sale
Plan:	remain after close	remain after close
Result:	achieve	block
Side+:		
Side-		block assure

Figure 5.1. Index for "Talked myself out of a sale"

having a different goal, to make the sale. In this story, the salesperson wants to have both the client's confidence and the sale.

All storytelling strategies generate triggering conditions in the same way. Recall that the triggering conditions are a generalization of the physical and social setting fields of the index. They are created by dropping those conditions that cannot be explained as necessary within the index. Of the features here, the sales relationship is obviously important, as is the fact that the salesperson is in the closing part of the sales call process, made evident by the salesperson's plan to remain after the sale has been closed. The trigger part of the RCD is therefore:

> WHEN the student is in the closing stage of the sales call, and
> talking to the decision-maker (the person in the sales relationship)...

In comparing a story against a storytelling situation, **Demonstrate risks** concentrates on the Actual part of the index, because it is trying to relate the story to events that are actually happening in the simulation. It tries to reconstruct the intentional chain found in the index. It looks for evidence that the student is engaged in a similar theme, has a similar goal and is using a similar plan to what is found in the Actual part of the index, but it looks for the student to succeed instead of fail. "Talked myself out of a sale" should get told if the student talks to the client after closing a sale but does not lose the sale as a result.

Evidence conditions for Demonstrate risks: Look for the student to have a similar goal and a similar plan, but to achieve an opposite result.

In "Talked myself out of a sale," the system knows that the student is going to have the seller theme in the sales closing setting, so no evidence need be gathered to recognize the student's theme if that context is present. The plan is also easily observable, since the system can readily tell if the student has remained after a sale is closed.

The RCD must also describe how to recognize something that is less obvious, that the student has the goal of making a sale. I have already described the difficulties in modeling a student's mental state, and recognizing every possible occurrence of this goal would certainly be difficult. SPIEL makes the simplifying assumption that the student really is playing the role assigned by the simulation. Given this assumption, actions of the student's can be taken as manifestations of mental states related to the performance of the task. If the student is going to visit a client with a prepared ad and other presentation materials, this is good evidence that the goal of making a sale is being pursued. SPIEL will be confused by a student who is trying to

Table 5.1. Impact types and their opposites

Impact type	Opposite impact
Achieve	Block
Serve	Threaten
Threaten	Serve
Block	Achieve

"psych out" the system or one who is trying to get every character angry just to see what will happen.

For this reasoning, SPIEL has a domain- and task-dependent library of *manifestation rules*. These rules enable it to reason about the mental states it is trying to recognize. They infer observable correlates of mental states that are likely to appear as the student acts in the simulated world. Here is one of the rules used in reasoning about the "make-sale" goal:

IF the setting is the closing, the decision-maker is present, and the goal is a sales goal,
    A POSSIBLE MANIFESTATION is the student telling the decision-maker the goal.

So to detect the goal of making the sale at the closing stage, SPIEL must look for the student asking the decision-maker for the order.

5.2.2. Implementing opposite-finding inference

If a similar theme, goal and plan can be identified, the **Demonstrate risks** strategy still needs to see that the student has achieved an outcome different from that found in the story: the student has succeeded where the story shows a failure. The **Demonstrate risks** strategy calls on SPIEL to predict an outcome that would be "very different" from losing the sale, different enough that it makes sense to tell the story as an alternative. Such opposite-finding inference occurs in each of the storytelling strategies discussed in this chapter.

An opposite outcome is not something that contradicts the original in every respect. It does not make sense to say that the opposite of achieving X is failing to achieve Y, where X and Y are opposite goals. If anything, these would be similar outcomes. An opposite result is an opposite impact on the same goal. As discussed in Chapter 3, there are four kinds of impacts found in results: a goal can be achieved, served, threatened and blocked. The impacts and their opposites are shown in table 5.1. SPIEL has opposite links between each impact and its opposite, and generates opposite results by creating an instance of the opposite impact for the same goal.

With these links, SPIEL can find the opposite of any given result. However, it must first determine what it needs an opposite of. In "Talked myself out of a sale", the actual result is the blockage of the sales goal and the desired result is the achievement of the same goal, so it is fairly easy to see that the different impacts on the goal of making a sale are crucial in this index. The opposite of the salesperson failing to achieve the sales goal is a salesperson (the student) achieving the sales goal, despite remaining with the client after the close. The completed RCD is therefore

WHEN the student is in the closing stage and speaking to someone who is the decision maker,
    LOOK FOR the student to ask for the order, and
        the student to remain after the sale, and
        the student not to lose the sale
    THEN TELL "Talked myself out of a sale"
        AS a "Demonstrate risks" story.

```
(defStrat DRK
      "Tell a story about a negative result of a particular course of
action when the student has just executed a similar course of action but
had success."
      :filter
      (and (:actual-result m-negative-result)
           (:agent m-student-role))
      :recognition
                ((:soc-setting :gen)
                 (:phys-setting :gen)
                 (:actual-theme :manif)
                 (:actual-goal :manif)
                 (:actual-plan :manif)
                 (:actual-result :result-opp :manif))
      :headline "A story showing the risks of your approach."
      :bridge "You <plan>.  Nothing bad happened but here is a story in
which doing that led to problems:"
      :coda "Nothing bad happened to you as a result of <plan>, but
sometimes <bad-result>."
      :vars ((plan :store :actual-plan)
             (bad-result :store :actual-result)))
```

Figure 5.2. Declarative representation for **Demonstrate risks** storytelling strategy.

The task of finding an opposite outcome is slightly more complicated when the main result in each column is the same. In this case, it tries to see if the columns can be differentiated on the basis of their side-effects.[2] In the "Long-term goals" story for example, the main result in the actual column is the same as that in the feared column. (See the index is shown in figure 5.4.) The real difference between columns arises in the side-effect slots. The feared side-effect is that the customer will not be sufficiently convinced of the ad's value; the positive side-effect that actually accrues is that the salesperson learns about the customer's long-term goals. For "Long-term goals," then, the opposite of the actual outcome is therefore not the student losing a sale (which would be the opposite of the main result) but the student failing to find out about the customer's long-term goals, the opposite of the positive side-effect.

> **Demonstrate risks:** Tell a story about a negative result of a particular course of action when the student has just executed a similar course of action but had success.

5.2.3. Representing storytelling strategies

SPIEL has a small number of inference types like the opposite result computation. These operations are combined to define strategies using a simple declarative representation. Figure 5.2 shows the definition for the **Demonstrate risks** strategy, containing the filtering conditions, the computation for creating recognition conditions, and the natural language templates. (Appendix B shows the representation of all the storytelling strategies and gives a full description of the representational vocabulary.)

The filter part shows the tests performed to verify that an index is compatible with the storytelling strategy. The tests are comparisons between slots in the index and particular MOPs. This strategy needs to see a story about a negative result happening to someone who is

---

[2]If there is no difference between the result, positive or negative side-effect slots in the two columns, SPIEL uses the result slot by default.

- Student argues for an increase in the customer's ad by showing him that he can get a co-op discount to pay for the increase.
- The customer instead decides that he should keep the same size, but use a co-op program to reduce his cost.

A story that shows an opportunity that didn't arise here.

You tried to sell a larger ad using co-op. It didn't work well, but here is a story in which that strategy worked.

This particular customer on pre-call gave some buying signs. He told me a little about his business. He was very "down" on Yellow Pages as a whole. He'd been in business for many years. I thought I interpreted some buying signs, but I really didn't. I went back to the office, and put together a nice program.

And upon the return visit, the customer totally changed. All of a sudden, he did not want to spend another penny. He did not need any more advertising. And that was that. There was nothing I could say that was going to change his mind about it. So what we did is we took a look at his ad. We took a look at the content of the ad, and started working on that at first. So I put him at ease, allowing him not to think that I was going to "try to get him into something that would work better for him." Then when I did produce the ad that I had put together for him, he again stated that he did not want to spend any more of his money. That was the ad that he's had in there for many years, seven years I think to be exact and that was that.

So I had to draw on some other ideas. I asked him what manufacturers he bought from, what manufacturers he bought the most from, and we started discussing co-op. It turned out that one of the companies that he bought a lot from had a very good co-op program when we looked in the co-op book. What we did is we figured out, based on his purchases, what amount of co-op he would get back from the program. We were then allowed to put color in the ad. Basically, the co-op paid for the color. He ended up with a white knock-out, much more powerful ad, totally-reworded. In his eyes, he didn't spend any money whatsoever, but it was a $135 increase.

You didn't have a good result from trying to sell a larger ad using co-op, but sometimes you can make a sale.

Example 5.2. Telling "Co-op success story" using the **Demonstrate opportunities** strategy.[3]

Demonstrate opportunities generates RCDs in a manner similar to that used in **Demonstrate risks**. The triggering conditions are generated again by examining the physical and social settings. There is not much in this index, because the salesperson was not very specific in his story. What is in the index is important: the sales relationship and the sales presentation, which is associated with the use of the co-operative discount argument.

To generate the evidence part of the recognition conditions, the strategy focuses on the Actual part of the index to match against the actions that the student takes in the simulation. It looks for a sequence of events that indicates the same intentional chain at work in the simulation as in the story, evidence that the student has the same goal, selling an increased advertising program, and is following a similar course of action. Then, like **Demonstrate risks**, it looks for an opposite result. This time, since the filter guarantees that the result is good, the opposite will be a poor outcome.

Evidence conditions for Demonstrate opportunities: Look for the student to have a similar goal and a similar plan, but to achieve an opposite result.

[3]Story from interview with Jim Beckett, video clip #18.

Physical setting:	presentation
Social setting:	sales target
Viewer:	seller
Perspective:	assumed
Agent:	seller
Anomaly Type:	plan

	Assumed	*Actual*
Theme:	seller	seller
Goal:	get increase	get increase
Plan:	sell benefits	use co-op
Result:	achieve	achieve
Side+:	achieve overcome $ objection	achieve overcome $ objection
Side-		

Figure 5.3. Index for "Co-op success story."

Again SPIEL must look to actions that can serve as manifestations of the student's goal. One good indicator of a student's selling intentions is the type of material she prepares to present to the client. The size and number of prospective ads that the student has prepared can be compared to the customer's current program to see if the student has the goal of selling a significantly large advertising program.

IF the setting is presentation and the goal is a sales goal,
A POSSIBLE MANIFESTATION is the presence of props that support that goal.

The idea behind this strategy is to show students stories with good outcomes when they have not done as well. To do this, the strategy locates an outcome opposite to the positive result shown in the index. This inference is done using the same opposite links used in **Demonstrate risks.**

In this index, the main result in both columns is the same, leading SPIEL to examine the side-effect slots to find distinguishing differences, but there are none: both columns have the same positive side-effects. Where the index gives no guidance, SPIEL uses the main result. The opposite of a goal being achieved is the blockage of the same goal, so the RCD looks for the student failing in the goal of selling an increased advertising program. The RCD for this opportunity looks something like this:

WHEN the student is in the presentation stage and speaking to someone who is the decision maker,
LOOK FOR the student to have prepared an ad program larger than
the customer's current one, and
the student to offer a co-operative discount, and
the student not to sell the increased program.
THEN TELL "Co-op success story"
AS a "Demonstrate opportunities" story.

Demonstrate opportunities: Tell a story about a positive result of a particular course of action when the student has just executed a similar course of action but had a poor result.

5.4. Demonstrate alternative plan

> Junior consultant: I just couldn't convince the client that they needed a full review of their business direction. They bought from our competitor who just offered them a system implementation.
>
> Experienced partner: I ran into a similar customer. When he downplayed the review, I didn't argue, just rewrote my presentation. Then after we got the job I did the review anyway and justified it. It was no problem.

This example shows the **Demonstrate alternative plan** strategy. The junior consultant has failed to achieve an important goal, using the best method she knows. The experienced partner has a story that shows success in a similar situation. It is like a **Demonstrate opportunities** story, but it is about an approach that is opposite from the one that failed. The partner agreed with the customer instead of arguing and the result was good.

The idea behind this strategy is to show students ways of doing things that they might not know about. The failure of a plan provides incentive for the student to learn some new way to accomplish the same goal. A tutor that could be assured that students would correctly understand their errors and learn from them would not need to intervene in this way, in keeping with the desire to have students learn from their own failures. However, in a complex domain, students who have failed may not have enough information to understand where they went wrong or to figure out how to do better. The junior consultant may not know any way to sell a job except by constructing an argument for each piece of work in the work plan. She might be tempted to "explain away" the failure by constructing the following type of explanation: "Arguing for the business review did not sell the job, but since I don't know any other way to convince the customer, I must not have tried hard enough." The radically-different approach taken by the partner challenges this view of how selling should be done. The partner does not believe in trying to hammer out every point, aiming for commitment first, then working out the details later.

It could be that the option shown in the story is one that the student considered but did not take. In this case, the student will have to explain whether her choice to reject the option was an error (since the approach worked in the story) or whether her choice was correct given that there was an important difference between the situation in the story and the one she faced. In either case, the story will contribute to the creation of an informed explanation of what went wrong and a greater understanding of the domain.

In example 5.3, the student asks a standard question about the customer's long-term goals, but doesn't do much with the answer. The customer's comment, even if joking, about wanting to fish and wanting his son to work more should trigger some questions in the salesperson's mind, such as the possibility of retirement. At the very least, the salesperson should try to get a serious answer to the question, but the student accepts the answer and walks out. In response, the storyteller shows the "Long-term goals" story about a salesperson in a similar situation who focused on a client's long-term goals, got him to spell them out in detail, and then used them as a selling point.

- Student asks about the customer's long-term goals.
- The customer says "To make my son work more, so I can go fishing."
- The student says good-bye and leaves.

A story about a different approach you might try

You gathered information about long-term goals in a cursory manner without much success. Here is a story about a similar situation in which an account executive used a different method and was successful.

Recently, I made a call to an electric contractor in Springfield, Ohio. One of the greatest things that a Yellow Pages person ever hears is "Leave it like it is," that person is a very good customer. But it's also a guy who needs a little incentive of what they should buy, or maybe even a little reinforcement as to why they're buying. This company is a small company owned by a father, his son and just one other partner, basically a three-person operation in a small town. They have a very large ad; they have a quarter page ad with four colors. At that classification it's probably the largest ad in the directory. As a matter of fact, I did a history of the ads from the past three years in that classification and their ad and every other ad in that classification had remained the same.

So when I got in there he had already told me that he wanted to leave it like it is, but he gave me the sense that he wasn't really sure why he kept buying it. So what I did was ask him flat out, I could sense that something was bothering him, I asked him. I said, "John, what's bothering you?" He said to me he wasn't sure that the cost justified itself. He wasn't sure that the ad was actually paying for itself. He opened up to me for the first time, after that. It set him at ease and we started talking about how much value a customer is to him, each call, and how many calls he needs to make that a profitable ad.

We also got into ideas of goals of his in the next five years, with his son coming up in the business and he getting close to retirement. We started thinking about how much more business he would have to do so he could obtain those goals. My advice in this situation is if you can get a customer to open up about the business, he's going to tell you a lot more about it from a goals standpoint in the next five years. This customer ended up buying the next larger size ad up to increase his business in electric contracting so he could achieve the goal of having the business strong enough for his son to take over in the next five years. He also, by talking to him, realized that the value of the product was so significant, to have in the book. Just from a dollars standpoint, it didn't take that many calls be an actual cost-effective program for him.

You gathered information about long-term goals in a cursory manner. It didn't work well. In the future, you might consider probing deeply into the customer's long-term goals.

Example 5.3. Telling "Long-term goals"

using the **Demonstrate alternative plan** strategy.[4]

5.4.1. Applying the strategy

The **Demonstrate alternative plan** strategy calls for the storyteller to tell stories about success. It is looking for stories in which the student analog achieves positive result; this is the same set of stories **Demonstrate opportunities** selects.

> **Filter for Demonstrate alternative plan:** Select stories whose indices show the student analog achieving a positive result.

The index for "Long-term goals" is shown in figure 5.4. The account is a difficult one: the customer has kept his advertising the same for many years. The salesperson fears that even if he manages to get the ad renewed, the customer will not be sufficiently convinced of the value

[4]Allan Beam, video clip #193.

Physical setting:	pre-call, service business, small
Social setting:	business partner, seller, father/son
Viewer:	salesperson
Perspective:	feared
Agent:	salesperson
Anomaly Type:	side-

	Feared	*Actual*
Theme:	seller	seller
Goal:	make sale	make sale
Plan:	accept renewal	probe long term goals
Result:	achieve	achieve
Side+:		know long term goals
Side-	customer not convinced	

Figure 5.4. Index for "Long-term goals."

of what he is buying. This could spell trouble for getting a sale next year. To sell the value of the Yellow Pages investment, the salesperson tries to get the customer to "open up" about the long-term goals for his business. Knowing the customer's perspective on his business gives the salesperson the necessary information to position the larger ad he wants to sell and to show how the ad program meets the customer's needs. The negative side-effect that was feared does not come to pass.

This index has a richer setting, but most of it can be generalized away. Nothing about probing the customer's long-term goals or the other parts of the Actual column have any specific connection to the father/son or business partner relationship. It would not make sense to restrict the story to be told only when the student is selling to a client who is known to have his son as a business partner. Note that this depends on the index in question. If there were another index for the same story that talked about the customer's need to expand his business so that his son could take it over, its recognition conditions would need to use the father/son relationship.

The derivation of the evidence conditions follows the pattern that we have seen, except that SPIEL applies its opposite-finding inference mechanism to both the plan and the result. The opposite result is needed for the same reason as in **Demonstrate opportunities**, to tell the story when the student has failed where the story shows success. The idea behind seeking an opposite plan is to look for a situation in which the student does something very different from what the salesperson did in the story.

As we have seen, SPIEL cannot be as sophisticated as a human storyteller would be in using such a strategy. SPIEL, with its limited knowledge of social relations, does not know the "right" way to approach a social situation, if there is such a thing. It starts from a plan that was shown in the story to be successful. Using the opposite-finding mechanism, it can find a plan that is radically different from this plan. There is a good chance that if the student is using such a radically-different plan and not succeeding, the opposite plan described in the story will show an option that the student did not consider trying or did not know about.

Evidence conditions for Demonstrate alternative plan: Look for the student to have a similar goal, but to use a very different plan and to achieve an opposite result.

In the example above, what constitutes an opposite plan from probing the customer's long-term goals? An obvious answer might be "not asking about long-term goals," but it is also true that, as in the example, a salesperson who delves briefly into the issue, but does not find anything useful, is still a good candidate to hear this story. These subtleties must be handled by the opposite-finding inference mechanism.

5.4.2. Inference implementation

The task of finding an opposite plan cannot be reduced to associating plans and their opposites as was possible for results. Suppose there is a plan to give a flashy sales presentation to a gatekeeper. Would an opposite of this plan be to give the same presentation to a different person, or would it be to give a non-flashy presentation to the same person? Both answers are plausible. The only way to distinguish between them is to look at the larger context, the index in which the plan is embedded.

As discussed earlier, in the simpler kind of inversion used for results, creating an opposite involves holding some features constant while inverting others. To compute an opposite plan, SPIEL starts with those features of plans that are dimensions along which opposites can be found. It selects among these possible dimensions by examining the *reference*, the plan from the other column of the index. The differences between the original plan and the reference are assumed to be the dimensions of importance for this particular index.[5]

The dimensions of opposition in plans have to do with impact that a plan has on resources held by the planner and others. Every plan has a primary resource act, resources that are used in its execution, and resources that are created through its execution. Creating an opposite plan requires that the resource act and the primary resource involved in that act remain constant. For the flashy sales presentation above, neither of the proposed opposites alters the resource that a sales presentation aims at creating (the sales contract) or the fact that the presentation is aimed at creating that contract. The opposites are the result of altering the manner in which the creation is performed: the target of persuasion or the type of persuasive message.

Once the dimensions of opposition have been selected, SPIEL creates a new instance with the same features as the original except for the features selected for opposition. They are given values that are opposite to what they contained in the original MOP. Here is a more formal statement of the algorithm:

Let M = the original MOP.

Let R = the reference MOP, if any.

To compute an opposite of M,

1. If there is an opposite link from M to a MOP O, return an instance of O.
2. Otherwise, let D = the dimensions of possible opposition associated with M.
3. If there is a reference R, let S = the subset of D for which M and R have differing values.
4. If there is no reference or if M and R do not differ, then retrieve the default opposition dimensions associated with M, and give S that value.
5. Create a new MOP, O. Let the slots of O = the slots of M except those slots in S.
6. For each slot s in S, get the slot value M_S from M.
7. If M_S is empty, then let $O_S = R_S$.
8. If M_S has a value, then call the inversion process recursively with M_S as the value and R_S as the reference. Let O_S = the result of that recursive call.
9. Return O.

---

[5]As with results, it is possible that there will be no difference between columns. For such cases, the system has default dimensions of opposition to fall back on.

Table 5.2. The resource slots of probe-long-term-goals and accept-renewal.

		Input MOP	Reference MOP
	MOP name	probe-long-term-goals	accept-renewal
Fixed slots	resource-act	creation	preservation
	resource-created	knowledge-of-long-term-goals	yp-sales-contract
Invertible slots	resource-protected		salespersons-time
	resource-used	salespersons-time, customers-time	min-salespersons-time, min-customers-time

Note that the definition is recursive. The system finds an opposite of a MOP by creating a MOP with opposite fillers in certain of its slots. Each of these opposite fillers must be derived using another opposite computation. Each of the fillers that participate in these computations must also have their possible dimensions of opposition defined so that their opposites can be computed. This recursive definition, of course, must end somewhere. Built-in inversion links like those found in results exist for certain kinds of values, such as values with ranges: high cost versus low cost, and for valenced values: positive emotion versus negative emotion. The recursive computation always bottoms out with these links.

SPIEL computes an opposite plan for the "Long term goals" story using this algorithm. The original MOP is probe-long-term-goals; the reference is accept-renewal. In comparing probe-long-term-goals and accept-renewal, SPIEL compares the resources the plans involve and the way they are employed, except for the fixed slots, resource-act and resource-created. There are eight possible dimensions of opposition associated with resource creation plans: resources used, resources protected, object of protection, resource deployment action, resource selection action, resource maintained, resource shared, object of sharing. Of these, the two plans differ with respect to resources used and resources protected. Therefore, an opposite of probe-long-term-goals, taking into account accept-renewal, is still going to be a creation plan for knowledge of long-term goals, so it will be an information-gathering plan, but it will differ in the resources used and the resources protected during its execution. Table 5.1 shows how the opposite mechanism views the two MOPs; figure 5.5 shows a fragment of SPIEL's memory containing them.

In the resource-used slot, the MOPs have different values, so SPIEL must call the opposite algorithm recursively. Before doing so, it must determine which of the two possible pairings of slot values to use. First, it looks to align resources based on their type. This does not help here because the fillers in both slots are all instances of the same kind of resource (time). SPIEL also looks to see if there are differences in control of the resources. This heuristic aligns salespersons-time and customers-time with min-salespersons-time and min-customers-time, respectively, because they are resources controlled by the salesperson and the customer.

Once an alignment is made, the opposite-finding function is called again with salespersons-time as the MOP and min-salespersons-time as the reference. Min-salespersons-time is a specialization of salespersons-time with the added constraint that the amount of time resource is small. The amount of the resource is unspecified

Table 5.3. The result of applying opposite-finding inference to `probe-long-term-goals` with `accept-renewal` as reference.

MOP name	`information-gathering-plan`
resource-act	`creation`
resource-used	`min-salespersons-time`, `min-customers-time`
resource-protected	`salespersons-time`
resource-created	`knowledge-of-long-term-goals`

in the `salespersons-time` MOP, so following step #7 in the algorithm, SPIEL uses the value from the reference as the value in the opposite. The result is therefore `salespersons-time` with amount = minimum, which is the same as `min-salespersons-time`. Similar reasoning holds for `customers-time`, where `min-customers-time` is both the reference and the result of the computation. In the resource-protected slot, there is also no comparable value in `probe-long-term-goals`, so the value from `accept-renewal` is considered to be opposite by default.

Table 5.2 shows the slots of the new instance returned by the opposite-finding inference procedure. As can be seen in figure 5.5, these features match most closely the existing MOP `cursory-information-gathering`. The new MOP becomes a specialization of this MOP with `knowledge-of-long-term-goals` in the resource-created slot. The plan to recognize, then, is the student carrying out cursory information gathering about the customer's long-term goals.

SPIEL also must generate an opposite result. In the index, the main results are the same: the achievement of the goal to make a sale. Differences between the columns show up in the side-effects slots. In the actual column, the positive side-effect is the salesperson learning about the customer's long-term goals. A result with an opposite impact would be one in which a salesperson fails to learn about a client's long-term goals.

The resulting RCD looks something like this:

WHEN the student is in the pre-call stage and speaking to someone who is the decision maker,
LOOK FOR the student to perform cursory information gathering
about the customer's long-term goals, and
the student to fail to find out the customer's long-term goals.
THEN TELL "Long-term goal"
AS a "Demonstrate alternative plan" story.

Demonstrate alternative plan: Tell a story about a successful plan to achieve a particular goal when the student has executed a different plan and failed to achieve the goal.

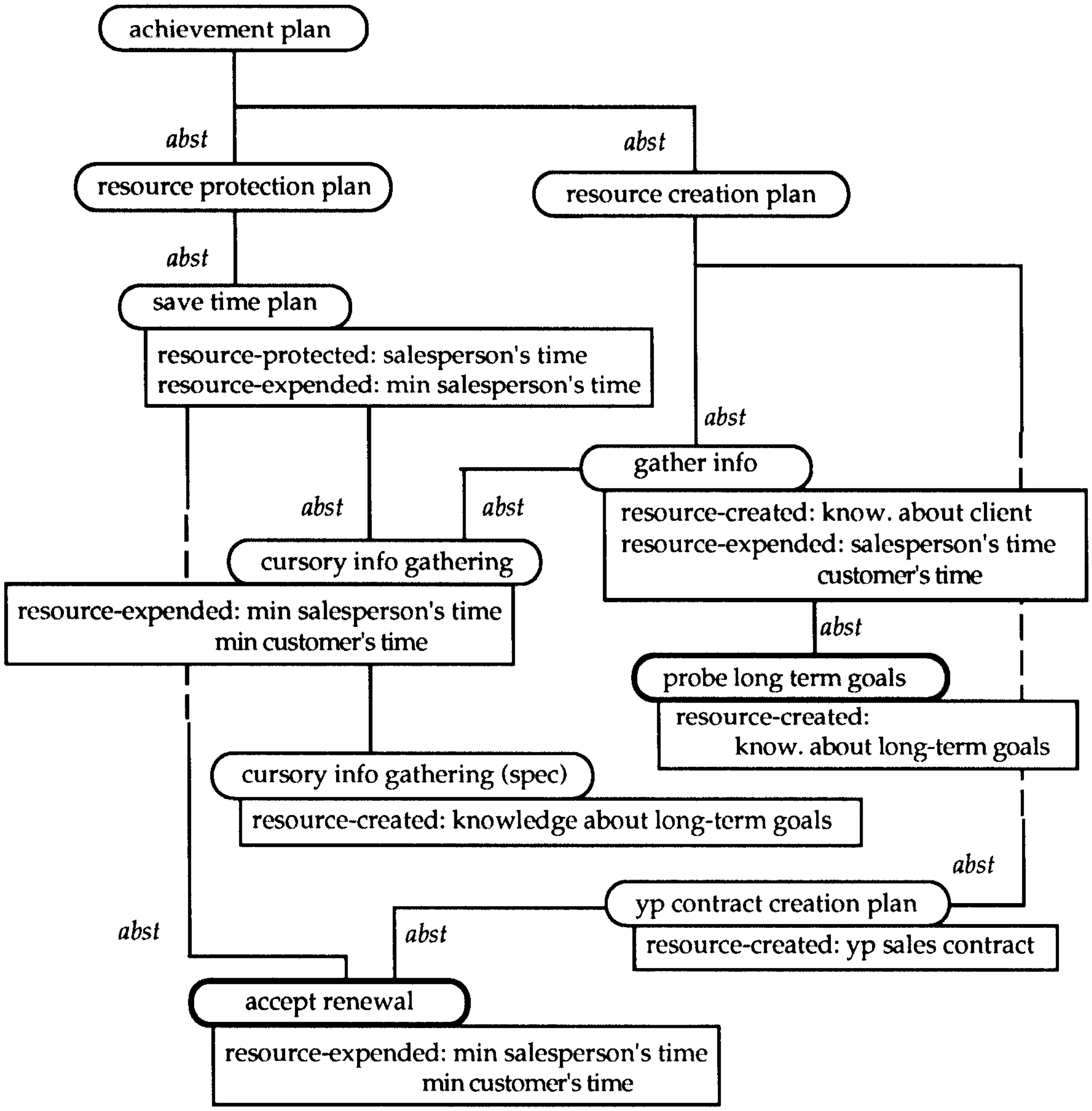

Figure 5.5. Fragment of SPIEL's memory used to compute the opposite of `probe-long-term-goals`.

5.5. Conclusion

The description of this set of storytelling strategies has brought up some important themes that will resurface as other strategies are discussed. All strategies use the physical and social setting of a story's index to create triggering conditions, performing simple explanation-based generalization where possible. Strategies also must reason about the observable manifestations of concepts in the index, constructing criteria for recognizing abstract concepts and using indirect criteria to detect students' intentions. An important feature of the indexing representation that these strategies exploit is the reference: a contrasting item from the other column of the index that helps refine what is important in an index.

What is unique about this set of strategies is the employment of the opposite-finding algorithm to compute alternate results and plans, so that stories can be used to show contrasts. SPIEL's alternative-finding computation involves knowledge about the allowable dimensions of difference and default dimension of contrast to use with different concepts, as well as opposite links built into SPIEL's knowledge base.

6. Stories That Critique Students' Expectations

6.1. Critiquing strategies

Bert: I think your parents will really like these tickets to Miss Saigon.

Rhonda: Don't count on it. I gave them tickets to Les Miserables, thinking they'd like it, but they left halfway through.

Like stories about failed goals, stories about failed expectations can be useful for teaching. In the example conversation, Bert tells Rhonda about his expectation, and she tells him a story about a similar expectation that she once had that turned out not to be met. He might reconsider whether his expectation is as unrealistic as the one in Rhonda had or if there is some difference between the two situations that makes it more likely that her parents will like Miss Saigon.

Such stories are important for teaching social expertise because newcomers may inappropriately carry over expectations from their social lives. A new salesperson may think, for example, that a friendly, talkative customer is more likely to buy than a "get down to business" type, when in many cases, the opposite is true (Craig and Kelley 1990). The anomalies in SPIEL's indices represent contrasts between expectation and reality, so SPIEL is well-poised to counter inappropriate expectations. There are three strategies that teach about expectations: **Warn about optimism**, **Warn about pessimism**, and **Warn about assumption**.[1]

These *perspective-oriented* strategies are inherently more difficult to implement than the strategies introduced in Chapter 5. The system has to make decisions based on what the student's perspective is likely to be. In the alternative-finding strategies shown so far, the problem of representing the student's thinking has not been severe, because the stories that these strategies tell show alternatives to observable features of the simulation. Their relevance can be fairly certain. However, for perspective-oriented strategies, the important features are aspects of the student's mental state such as hopes or fears. SPIEL must be able to recognize them. The problem of recognizing a student's state of knowledge is known as the problem of *student modeling* in intelligent tutoring systems (ITS) research.

The approach to student modeling that has traditionally been used in ITS research is the construction of an operational model (vanLehn 1988), a model that is sufficiently detailed that it accounts for the whole of the student's behavior. An operational model is often a set of rules that would solve a problem in the same way that a student does. A student's mental state at any point can be computed by an accurate model that processes the same problem as the student in the same way.

This type of model is not easily extended to the kind of knowledge that SPIEL needs to capture. ITSs are designed to handle students who are complete novices. A model that captures all of what a student knows about an area is only possible when most of that knowledge comes from experience the student has with the tutor (vanLehn 1988). A complex social skill, such as selling, is refined from the social expertise the student already has. Students have to learn which of their existing skills contribute to the new area and which detract. They learn new concepts and new skills, and fit both into their existing repertoire. Further, as a student becomes an expert, domain knowledge and experience are melded with

---

[1] Of course, some indices represent, not the perspective of the student analog, but that of a person in another role, such as the customer in sales stories. The storytelling strategy **Explain other's perspective** described in Chapter 8 uses such stories to teach about the viewpoints of others.

existing abilities into a distinct personal style. Adult students do not begin learning a social skill as complete novices. They bring to bear a great deal of prior knowledge and experience.

Some intelligent tutoring systems are able to handle students who bring with them a pre-existing understanding of the concepts (Burton 1982). They do so by building their student models from observation of the student's performance in the task environment. In a social domain, the prior knowledge and experience that students have is so large that a tutor will not be able to discern the state of the student's knowledge merely by observing a brief interaction with a simulated social world. It is unlikely that a model developed in this manner will successfully account for the individual idiosyncrasies in students' knowledge of and experiences in the social world.

SPIEL cannot have an operational model, one that represents the whole of a student's social reasoning. For social skills, it is unlikely that even human instructors have models at this level of detail. Suppose Rhonda, instead of having a conversation with Bert, were just observing him buy the theater tickets. It would be hard to tell if he were planning to give them as a gift to her parents, and it would be pretty annoying if she were to intervene with her story if he did not have that in mind.[2] The teacher is in a similar position when observing a student.

Rhonda does not need a full model of Bert's thinking in order to determine the relevance of her story. She could make more observations. She might see that Bert had already done some preparation such as buying an anniversary card, or comparing the dates of performances against the dates of her parents' vacation. Such observations would give her more confidence that she was intervening appropriately. In addition, before telling the story, she might confirm her suspicion by asking Bert directly: "Are you thinking about giving these to my parents?"

SPIEL does both of these things when telling stories intended to address the student's perspective. As part of its presentation, it prefaces the story with a question, saying essentially "If you are expecting X, then I have a story for you..." and it tries to identify preparations that the student might make that characteristically indicate an expectation.

Before addressing these strategies, it is worth noting that each has two variants. The student may have an expectation that is directed towards someone else, such as Bert's expectation about Rhonda's parents, or an expectation about his or her own action, like a salesperson who believes that she will be able to convince the customer of something. A different type of story is called for in these two cases. For example, in the "Office manager only taking orders" story, which will be seen shortly, the salesperson hopes that the office manager will listen to his presentation and report that material back to the owner. That hope is not realized. This story should be told if the student thinks that someone in a similar position in the simulation will help in this way. I call this the *other-focused* variant of the strategy since the expectation is about someone else. By contrast, in the "$19/month is too much" story (also below), the salesperson hopes to achieve something, namely selling a display ad to a new customer. It gets told when the student is similarly optimistic. This is the *student-focused* variant, because it is directed towards the expectations students have of themselves. Each variant of the strategies is described in the sections that follow.

6.2. Warn about optimism

The example that started this chapter is an example of the strategy **Warn about optimism**. Rhonda's story is about a hope that she had that was not realized: the hope that her parents would enjoy a show she gave them tickets to. She tells the story when it appears that Bert has a similar hope in order to get him to rethink this expectation. Such a story will be helpful to a

---

[2]Of course, this does happen in real life, but a tutor should try to avoid nagging, if possible.

- Salesperson is dealing with a customer that is a small new business.
- Salesperson designs a large-sized display ad.

A warning about something you just did.

If you are hoping that you will sell [the customer] a large ad, you may be disappointed. Here is a story in which an account executive had similar hopes that were not realized.

Five years ago in Indianapolis, I called on a florist. It was a lady and her sister. It was a brand new customer. I went out. It was in their home. They had gotten a business phone. They wanted a Yellow Page ad. Being new, they were skeptical. They didn't know much about advertising; they had just started. I told them the price of an ad and they were shocked. It was small display ad. Just shocked. Couldn't believe that we could charge $100 for a display ad. I settled that year for a 1/2 inch ad that cost, I believe, $19 a month and they were shocked at the $19/month. I walked out of there. They were two nice ladies, young housewives trying to start a business. And I kind of laughed to myself and thought, "You know, 1/2 inch ad, $19/month. I hope they make it. They probably won't be here next year." I went back the next year. They were still there. They had received some calls, not many, but they were growing. This time I sold an inch and a half ad with color and they went to like $70/month from $19 and were very pleased with me and the Yellow Pages.

The next year, she had moved to a storefront. The sister had left her and she was now the sole owner and she had a location on Michigan St. This year she bought an 1/8 of a page display ad, went to about $300/month. Each year it went on. This particular Indie that I finished last week. She is now with our new white knockout, color, 1/4 page, and she's spending approximately $1300/month: a very, very strong Yellow Page believer. In fact, she laughed with me this year because she recalled that first year how she thought that $19 a month was so expensive, "And here I am spending $1300 a month." She said, "I don't believe it, but it works."

A hope that you will sell a new customer a large ad may be unrealistic.

Example 6.1. Telling "$19/month is too much" using the strategy **Warn about optimism**, student-focused.[3]

student who is learning what sorts of expectations to have. It shows that it may be unreasonable to aim too high, and counteracts the student's overly-optimistic desires with a story that shows how a similar desire was not realized.

6.2.1. Applying the student-focused version of the strategy

Consider the student-focused variety of this strategy. That means telling a story about a salesperson's desire to accomplish something, as in the situation in example□6.1. The customer has a new business and is starting out small. The student designs large display ads that would be more appropriate for a larger business. The storyteller intervenes with the "$19/month is too much" story about a salesperson who hoped to sell a display ad to a small startup business but found the owner could only afford a small ad at first. The storyteller draws a parallel between the salesperson's desire to sell a display ad in the story to the student's similar intention, evidenced by the preparation of the ad.

What must this strategy look for in a story? Obviously the story must contrast something that was hoped for against something different that actually transpired. Unlike the alternative-seeking strategies, **Warn about optimism** is not concerned with the outcome of the story whether successful or not. As long as there is an unmet expectation, a storyteller can tell the story when the student gives evidence of a similar expectation. In this story, the salesperson does make a sale, but since he does so with a plan different from the one he had hoped to use, the story serves as a warning about that hope.

---

[3]Gant, video clip #257.

Physical Setting:	presentation, retail, new business
Social Setting:	sales-target
Viewer:	salesperson
Perspective:	wanted
Agent:	salesperson.
Anomaly Type:	plan

	Wanted	*Actual*
Theme:	seller	seller
Goal:	make-sale	make-sale
Plan:	propose-display-ad	size-ad-for-budget
Result:	achieve	achieve
Side+:		
Side-		

Figure 6.1. Index for "$19/month is too much"

Filter for Warn about optimism, student-focused: Select stories whose indices show a contrast between something the student analog wanted to do and what actually happened.

The index for "$19/month is too much" is shown in figure 6.1. The index has a plan anomaly: the salesperson had hoped that the plan of presenting a display ad would help make the sale, but the plan that actually worked was to put together an ad that was sized appropriately for the client's budget.

For **Warn about optimism** and the other perspective-oriented strategies, the anomaly is the focus of evidence gathering. This sets them apart from the alternative-finding indices that look for a chain of events following the intentional chain in the Actual part of the index. These strategies address actions and events that have actually occurred and can be observed.

Perspective-oriented strategies do not use the whole intentional chain in generating evidence conditions. Trying to identify a single mental state in the student is difficult; requiring a match against a whole series of such states would make these strategies unusable. In "$19/month is too much" the salesperson wants to make a sale and do so using the plan of proposing a display ad. This desire is thwarted by the economic circumstances of the florist. This anomaly, the contrast between the hopes of the salesperson and the reality, is what this strategy seeks to address. It is trying to find an expectation on the part of the student that is likely to fail in the same way as the expectation of the student analog in the story.

SPIEL is attempting to recognize an expectation on the part of the student that corresponds to an anomalous expectation in the story. This expectation is of the form: "the student expects X rather than Y," where X is a slot in the Perspective column of the index and Y is the comparator from the Actual column.

Evidence conditions for Warn about optimism: Look for the student to give indications of having the same anomalous desire as the student analog.

One way that SPIEL tries to recognize the student's intentions is through *preparatory actions*. SPIEL assumes that a student who is faithfully playing the salesperson would prepare an ad only if he or she hoped to sell it. Preparatory actions are often part of the skill the student is learning. For example, a Yellow Pages salesperson is taught to prepare a presentation by putting together a graphical aid called a pitchbook before meeting with a client to close a sale. Actions taken in preparation tell the system what the student is preparing to do, and, possibly, what expectations are present.

An expectation will manifest itself in different ways in different contexts. In a pre-call context, the desire to propose a display ad may be manifested by the student through discussion of the possibility of display advertising for the customer. In the preparation context, the student will manifest a desire to sell a display ad by preparing such an ad.

6.2.2. Inferring triggering conditions

I have briefly covered the generalization of a story's setting that goes into the creation of triggering conditions. This example requires that we go into the process in greater depth. The plan in the Actual column, sizing advertising according to the customer's budgetary requirements, does relate to some aspects of the setting, such as the fact that it is a small business and that it is new, but drawing this connection requires actually doing a certain amount of additional explanation.

SPIEL reasons about the customer characteristics that are important to particular goals and plans through an analysis of the resources involved in a particular goal or plan and the kind of action performed on that resource. There are two ways that a plan or goal may have such an impact. The plan or goal may affect some resource that is social, shared between two actors, such as the time consumed by a conversation. Or it may involve resource exchange between the actors. SPIEL examines the resource constraints that are implied by a plan by using a set of *contextual analysis* rules. In reasoning about the plan "size ad for budget," SPIEL uses the rule

> IF there is a constraint placed on a resource expenditure in a plan, THEN assume the constraint holds for the actor of the plan.

In any ad proposal, the salesperson is constructing a proposal that consumes a certain amount of the customer's money, but in "size ad for budget," the salesperson is explicitly factoring the monetary resources available to the customer into the creation of the proposal. A customer characteristic associated with such a restriction is a cost constraint on the part of the customer.

SPIEL infers that the client has a restricted availability of money, then tries to identify features of the setting related to that characteristic. SPIEL surveys the features of the social and physical settings of the index using the restrictions it has found. It rules in features important to the telling of the story and rules out unimportant ones, generalizing the storytelling context. To deal with restrictions, SPIEL has *restriction stereotypes.* The stereotypes employed in this example are

> Small businesses often have limited financial resources.
>
> New businesses often have limited financial resources.

With these rules, SPIEL can determine that cost-consciousness is related to two of the features of the index's setting: the size of the business and the fact that it is new. These setting features therefore become part of the triggering conditions in the recognition condition description.

Recall the importance of preparation in recognizing the student's intentions. To tell "$19/month is too much" using the student-focused variant of **Warn about optimism**, SPIEL needs to create a trigger that will recognize a context in which it might be possible to recognize students making preparations that indicate their intentions. The structure of the selling task makes it possible to do this. Each stage of the sales call has one or more prior stages in which the salesperson prepares for the actions at that stage. Therefore the *presentation*, the point at which the salesperson hopes to make his display ad recommendation in the story, has as its preparatory stage the period during which the salesperson designs a prospective ad and puts together a sales presentation, which is the *preparation* stage, and also the *pre-call* stage where the student gathers information for arguments that will be made in the presentation. The whole set of preparatory connections is shown in table 6.1. Using this knowledge, the system builds two triggers for the "$19/month is too much" story: one that

Table 6.1. Sales call stages and their preparatory precursors

Sales call stage	Preparatory stage(s)
account review	N/A
pre-call	account review
preparation	pre-call
presentation	pre-call, preparation
close	pre-call, presentation

looks for an opportunity to tell the story during the pre-call and one that looks during the preparation stage.

6.2.3. Inferring evidence conditions

For the evidence part of the RCD, SPIEL is looking for ways to recognize that the student is hoping to propose a display ad rather than pay attention to the client's budget. In a pre-call context, the student is carrying on a conversation with the client, so preparation for proposing a display ad will be evidenced by discussion of that kind of advertising. In the context of ad preparation, the student will show an intention to sell a display ad by constructing it and including it in the sales presentation. SPIEL has manifestation rules that handle these cases:

IF the context is the pre-call and evidence is needed about a property of a proposal, A POSSIBLE MANIFESTATION is any discussion of that property during the pre-call.

IF the context is preparation and evidence is needed about a property of a proposal, A POSSIBLE MANIFESTATION is any proposal with that property.

The system must also gather evidence that the student does not have the "correct" expectation, one that is in line with what actually happened in the story. The system knows that the only precondition of the "size ad for budget" plan that can be satisfied in the pre-call context is the gathering of information about the client's budget. If the student does not do this, that is good evidence that the student is not using the "size ad for budget" plan. Similarly, while preparing an ad, the distinction in the anomaly is cost, so if the student does not prepare a cheaper ad than the display ad, this indicates no consideration of the client's budget. *Manifestation restriction rules* are employed to place these restrictions on the recognition conditions:

IF the context is the pre-call and evidence is needed about the absence of preparation for a plan, A RESTRICTION ON THE MANIFESTATION is that the student should not satisfy any of the preconditions of the plan.

IF the context is preparation and evidence is needed about the absence of a property from a proposal and a contrasting proposal has more of that property, A RESTRICTION ON THE MANIFESTATION is that no proposal should have less of the property than the contrasting proposal.

Two triggering conditions were generated for this tutorial opportunity. Since triggering conditions recognize a particular context and the evidence conditions depend on this context, separate RCDs are required for each possible context. Therefore, there are two RCDs required for this combination of story and strategy:

WHEN the student is in the preparation stage and
the client is someone who is new in business and
the client has a small business,

LOOK FOR the student to prepare a display ad, and
the student not to prepare a smaller ad,
THEN TELL "$19/month is too much"
AS a "Warn about optimism" story, student-focused.

WHEN the student is in the pre-call stage, and
the client is someone who is new in business and
the client has a small business,
LOOK FOR the student to tell them about the benefits of
a display ad, and
the student to fail to ask about the client's advertising budget,
THEN TELL "$19/month is too much"
AS a "Warn about optimism" story, student-focused.

6.2.4. Applying the other-focused version of the strategy

The other-focused version of the **Warn about optimism** strategy is used when a story is about what the student analog hoped someone else would do. This is more in line with the example of Bert and Rhonda. Bert hopes that Rhonda's parents will enjoy the theater production.

Using similar types of reasoning to those seen above, SPIEL works from what someone hoped for in a story to infer actions that might indicate that the student has the same hope. In example 6.2, the student clearly wants the office manager to act as a go-between to get to the real decision-maker. The story shows an instance in which the intermediary acted more as a road-block than a go-between, suggesting that the student may want to revise this expectation.

Filter for Warn about optimism, other focused: Select stories whose indices show a contrast between what the student analog wanted someone else to do and what the person actually did.

6.2.5. Generating the recognition condition description

As we have seen, perspective-oriented strategies work with the anomaly part of the index. The index for "Office manager only taking orders" is shown in figure 6.2. The anomaly is in the plan that the gatekeeper is expected to follow. The salesperson wanted the office manager to communicate information from the sales presentation to the owner, but the gatekeeper remained impervious to the sales arguments, sticking to the owner's original decision.

A look at the anomaly shows that the employer/employee relationship must be important since the desired plan is one of communication between employee and employer. This (and the presentation stage) becomes the trigger.

Recall that the student-focused variant of a perspective-oriented strategy is looking for students' expectation about their own actions, using the fact that students will often prepare themselves. The other-focused variant can use this approach to look for the student doing some kind of counter-planning such as building a presentation around anticipated objections. Usually, however, other-focused strategies must look for the evidence of a student's expectations about a person by watching how the student interacts with that person.

What would be evidence that the salesperson expects an employee to communicate sales information to a decision-maker? The purpose of a sales presentation is to convince a decision-maker to buy. If a student gives a sales presentation to someone other than the decision-maker, it means either that the student has been misled about who the decision-maker is or that the student wants the sales information to be recommunicated to the decision-maker. The following manifestation rule captures this insight:

• The student is finding it difficult to get through to the lawyer in a law firm and instead makes an appointment with the office manager.
• In the presentation, she gives the office manager detailed facts and figures to support her recommendation.

A warning about something you just did

If you are hoping that [the office manager] will pass your presentation information on to the buyer, you may be disappointed. Here is a story in which an account executive had a similar hope that was not realized.

I once had an automobile dealer that had a very large program at the time I went in. I was a little bit nervous about going in and seeing this customer the second year after I had increased the business around 3200 dollars the prior year. That's always a nervous situation because you're worried about how well it worked, what the customer thought of it, mainly what the customer thought of the program. I went in to the customer's office and I was told that I would not be dealing with the owner anymore that I would be dealing with the general sales manager. That was kind of a shock to me and I saw that as something that wasn't really too great. It made me think that the owner didn't want to see me about this account anymore because he had made the decision last year and possibly thought he had made a bad decision.

My feelings were true; they were correct. I went back and saw the customer; it was the general sales manager. He told me that they wanted to take the majority of the advertising out. Keeping in mind that auto dealers kind of do that. They will jump from one advertising arena into another with some frequency. They have a lot of dollars to invest and this is just the way they do business. So for them it wasn't anything out of the ordinary to place an order with one company and to not place it the next year and to place it with somebody new.

But to me, it was something totally different. I didn't really see it that way. So it was kind of upsetting. So I did go back and ask my manager if he would ride with me on the account. We spent quite a bit of time with the customer going through every point that we could think of for him to keep the advertising in the directories. It ended up the sales manager was taking his orders from the owner and was more or less powerless to change what the owner decided to do. So it ended up being around a 3500 dollar decrease. It was definitely one of my most "non-memorable" experiences.

A hope that a gatekeeper will pass your presentation information on to the buyer may be unrealistic.

Example 6.2. Telling "Office manager only taking orders" using the strategy **Warn about optimism, other-focused.**[4]

IF the expectation to recognize is the expectation of a resource transfer of some resource from the student by person with role A to person with role B, then
A POSSIBLE MANIFESTATION is the transfer of that resource from the student to person with role A combined with knowledge on the student's part of that person's role.

Applying this rule to the index and combining with the trigger derived earlier yields the following RCD:

WHEN the student is in the presentation stage, and
the student is talking to an employee of the decision-maker,
LOOK FOR the student to know the employee is not decision-maker,
the student to give a sales presentation to the employee,
THEN TELL "Office manager only taking orders"
AS a "Warn about optimism" story, other-focused.

[4]Beckett, video clip #40.

Physical Setting:	presentation, large, retail
Social Setting:	sales target, employer/employee
Viewer:	seller
Perspective:	wanted
Agent:	manager
Anomaly Type:	plan

	Wanted	*Actual*
Theme:	go-between theme	gate-keeper theme
Goal:	purchase-ad	implement-decision
Plan:	communicate presentation to boss	ignore presentation
Result:	achieve	achieve
Side+:	serve-make-sale	
Side-		block-make-sale

Figure 6.2. Index for "Office manager only taking orders" story

Warn about optimism: Tell a story about a desire that a salesperson had which was not realized when the student appears have a similar desire.

6.3. Warn about pessimism

Leona: I'm really dreading this blind date. He's sure to be a nerd. Why did I let you talk me into this?

Maxine: I remember being pretty down on blind dates too, just before I met Norman on one. We've been together 3 years now.

Students' unrealistic hopes will often correct themselves because they will fail to be realized. A more subtle but equally insidious problem is a tendency to aim too low, to be dissuaded by unrealistic fears. Student who aim too low may never discover that they could have done better. It is important therefore to confront such fears with stories of others in a similar position with similar fears that were not realized.

The conversation shows the **Warn about pessimism** strategy at work. Leona expresses a fear of blind dates. Maxine counters by telling a story about a time when she had a similar fear that was not borne out. The story is an attempt to show Leona that perhaps she is being too pessimistic since not all blind dates are disasters.

Countering low expectations is particularly important in teaching selling. There is a saying in sales that somebody gets sold in every sales encounter: either the customer is sold the product or the salesperson is "sold" on the idea that the customer doesn't want it. Salespeople with unrealistic fears about the customer's resistance sell themselves on defeat before the sales call even takes place.

Warn about pessimism is a perspective-oriented strategy that shows that the world may be more friendly than the student thinks. If the program has evidence that the student fears to do something, it may be able tell a story about a time when that fear was not realized or when it was worked around. See example 6.3.

- The student doesn't do much research before talking to an attorney.
- The attorney has a large ad in a different book, but has an in-line listing in the book being sold.
- The student doesn't know this and makes a small ad as a recommendation.

A hint about something you just did

If you fear that you will propose too large an increase, you may be surprised. Here is a story in which an account executive had a similar fear that was not realized.

I know one customer I had which is a podiatrist. I wanted him to increase the ad size, and of course, the dollars, he was always talking dollars. But I went back out there with—not a big increase—but a nice small increase, getting him to the next size. He bought everything, hook, line and sinker and I know now that I would have just, I'm not saying push him, but if I would have really done exactly what I felt rather than thinking in the back of my mind, the 'dollar factor.' I think he would have bought the larger ad as well.

So it's kind of given me a little lesson too that not to go overboard but don't get taken by the fact that they don't want to spend any money or leave it like it is, because nobody's going to have you walk in there and say "Hey, I'm waiting to buy something." It just doesn't happen. I learned my lesson with that one.

A fear that you will propose too large an increase may be unrealistic.

Example 6.3. Telling "Aimed too low" using the strategy **Warn about pessimism**, student-focused.[5]

6.3.1 Applying the student-focused version of the strategy

Warn about pessimism looks for stories that contrast something that a salesperson fears to do or achieve against a reality whose outcome was positive. In this strategy, the outcome is important. If Leona dreads her date will be a nerd, Maxine will not be very encouraging if she says "I once thought a date would be a nerd, but he turned out to be a sleazeball." The outcome in the story must be positive for the story to have a chance of acting as a positive example.

In "Aimed too low," the salesperson feared that proposing a large ad would not have a good result, so she proposed a smaller increase and was successful, not however achieving her original goal of getting a large increase and not giving the client the ad program she thought was best.

> **Filter for Warn about pessimism, student-focused:** Select stories whose indices show student analogs with something they feared to do turning out to have a good outcome.

Similar again to **Warn about optimism**, this strategy looks for preparatory actions that could indicate the existence of a certain fear. Fears show up in an opposite manner from hopes. Students manifest hopes by having lofty aims and not anticipating any problems in carrying out their plans. A student who is fearful will shrink from reasonable goals and be overly conservative in planning.

> **Evidence conditions for Warn about pessimism:** Look for the student to give indications of having the same anomalous fear as the student analog.

SPIEL does not really know what is a reasonable goal in any given situation, but it can look for the student adopting a goal and backing down from it. To find evidence for a similar "backing down," we first have to see that the student has the original goal but then demonstrates a change in direction. That original goal may be expressed explicitly by the student, in bringing

[5]From interview with Barbara Knisley, video clip #327.

Physical setting:	presentation, professional, small
Social setting:	sales-target
Viewer:	salesperson
Perspective:	feared
Agent:	salesperson
Anomaly Type:	plan

	Feared	*Actual*
Theme:	seller	seller
Goal:	get-increase	make-sale
Plan:	propose-big-increase	propose-small-increase
Result:	block	achieve
Side+:		
Side-	block-make-sale	block make sale best for client

Figure 6.3. Index for "Aimed too low"

up the question of increasing the ad size when talking to the customer, or it may show up implicitly in the preparation process, if the student prepares a larger ad and then changes his or her mind.

A slightly more intelligent SPIEL might be able to go one step farther and recognize that the situation called for the student to propose a major increase in advertising, and see that the student failed to aim for this. However, to get a storyteller to have this capacity would involve all of the problems of modeling expert knowledge of selling. Even experts disagree about what factors are legitimate ones for reasoning about selling opportunities and how they should be weighted, due porbably to issues of personal style or local culture, Lesgold's *dialects of expertise* (Lesgold and Lajoie 1991). Even if such an expert model did exist, SPIEL would have to monitor the student's information gathering enough to determine that the student had gathered all of the information needed to draw the correct conclusion. It would also have to rule out the possibility that the student simply did not understand the evidence that the circumstances called for a major advertising increase. Even a slight improvement in SPIEL's employment of this strategy would entail an extensive commitment to modeling.

6.3.2. Inference implementation

The index is shown in figure 6.3. The salesperson feared that the customer would object to the cost of the ad that she thought was needed, so she reduced her aims, failing, however, to give the client the best possible ad program. The action takes place in the presentation, so SPIEL looks for preparation that might happen during pre-call and preparation. The anomaly points to a backing down from selling a large ad to selling something smaller. The feared plan demands more monetary resources on the part of the client, which suggests the "small business" part of the setting is important to keep.

SPIEL identifies the evidence needed to indicate that a student has the same fear as in the story, looking for the original intention and detecting a "backing away" from that intention. The fear in the anomaly is directed toward the plan of proposing a large increase in the ad. So, SPIEL looks for the student showing intention of aiming for a large increase, and then backing away from this idea. The change of heart can be manifested in two ways: the student may tell the client "Forget what I said about the increase," or simply not put together any large ads.

> IF the context is pre-call and evidence is needed of the student backing down from a proposal with a certain property,
> A POSSIBLE MANIFESTATION is the student downplaying the importance of that property.

IF the context is preparation and evidence is needed of the student backing down from a proposal with a certain property,
A POSSIBLE MANIFESTATION is the creation of a proposal without the property with the restriction on the manifestation that any proposals with the property not be in the presentation materials.

Because of the possibility of recognizing the tutorial opportunity in two different contexts, there are two RCDs. One looks for the student to talk about the need for increased advertising, but in the face of resistance, either to tell the customer to forget it or when preparing ads, not to create any that represent a substantial increase. The second operates entirely within the preparation phase, looking for the student to prepare an ad that is a large increase, but then reconsider.

WHEN the student is in the pre-call stage, and
the client has a small business.
LOOK FOR the student to mention the possibility of
increasing the ad's size, and
EITHER to tell the client an increase in not important, or
the student to enter the preparation phase, and
prepare an ad that is not a substantial increase,
THEN TELL "Aimed too low"
AS a "Warn about pessimism" story, student-focused.

WHEN the student is in the preparation stage, and
the client has a small business.
LOOK FOR student to prepare an ad that is a substantial increase, and
prepare an ad that is not a substantial increase, and
not include the larger ad in the sales presentation.
THEN TELL "Aimed too low"
AS a "Warn about pessimism" story, student-focused.

6.3.3. Applying the other-focused version of the strategy

Warn about pessimism can also work in an other-focused manner to address unrealistic fears the student might have about what others in a sales situation may be doing or thinking. In example 6.4, the student appears to avoid discussion that involves specific industry terminology. The story is about a salesperson who feared that her lack of knowledge about the client's business would threaten her prospects for a sale, but who discovered that it did not. The story shows the student that ignorance of the client's business is not necessarily an insurmountable difficulty.

This storytelling strategy looks for stories that portray a fear on the part of a salesperson that is not realized by the events of the story. In the "Taxidermist" story, the salesperson is afraid that her ignorance will be a turn-off to the professional taxidermist, but he ends up buying anyway. The strategy calls upon SPIEL to look for evidence of a similar fear on the part of the student and to tell the story to point out that the fear may be unrealistic.

Filter for Warn about pessimism, other-focused: Select stories whose indices show a contrast between something the student analog feared someone else would do and what actually happened.

- Client uses industry terminology.
- Student changes the subject.

A hint about something you just did.

If you fear that Mr. Swain expects you to be completely knowledgable, you may be surprised. Here is a story in which an account executive had similar fears that were not realized.

One time I was talking with an older gentleman about his taxidermy business. I didn't really know too much about hunting or fishing or any of those kind of outdoor things. So I was having a little bit of trouble. I ended up discussing his business with him enough so that he would tell me important things about why his taxidermy business was different from someone else's. The whole time I was talking with him I didn't think he was going to buy anything because I was so clueless about what he was talking about. But he did end up talking enough about his business that he decided he wanted to tell everybody about all these things that he had. When he got excited about wanting people to know it, he agreed that the Yellow Pages was the best medium to get it out.

The fear that a client expects you to be completely knowledgable may be unrealistic.

Example 6.4. Telling the "Taxidermist" story using the storytelling strategy **Warn about pessimism, other-focused.**[6]

The index for "Taxidermist" appears in figure 6.4. The main anomaly, the principle thing that the salesperson feared, is a negative side-effect. The salesperson feared that there would be a bad side-effect of the customer's desire for a salesperson who understood his business, namely that she would not make the sale. In actuality, there was not a negative side-effect, but a positive one of the sale being made.

6.3.4. Inference implementation

This index shows a case where the anomaly that the indexer has chosen as the main anomaly does not carry much information. The student is always going to fear losing the sale: that comes with the sales territory, and there is nothing useful in the actual column to contrast it with. That does not mean the story was indexed badly: that *is* the principal thing the salesperson fears. What it means is that SPIEL must search the index for the reason why this fear was particularly salient in this story.

It traces back through the two intentional chains to find the first point at which there is a discrepancy that might account for the anomalous outcome. It finds that motivating discrepancy in the goal slot. The customer's desire to have an ad turned out to be stronger than his desire for a salesperson that he didn't have to explain everything to.[7]

This process is called *anomaly derivation search*. It occurs whenever the anomaly slots contain a distinction that is trivially true. The result of the anomaly derivation search on these indices is the contrast between the taxidermist's goal of having a knowledgeable salesperson and his goal of getting a Yellow Pages ad. SPIEL has to recognize a situation in which a student fears that the client's desire for a knowledgeable salesperson is stronger than the desire to buy an ad. In other words, a case where a student fears that looking ignorant could cost the sale. The following manifestation rule is relevant here:

> IF the fear is that the customer will expect a certain property of the salesperson, A POSSIBLE MANIFESTATION of that fear is the student avoiding an opportunity to display that property.

---

[6]From interview with Amber McLean, video clip #309.

[7]Note that it would be incorrect to say that this slot is the anomaly, because the client has both goals, not one rather than the other.

Physical Setting:	pre-call, small, service
Social Setting:	sales-target
Viewer:	salesperson.
Perspective:	feared
Agent:	taxidermist
Anomaly Type:	side-

	Feared	*Actual*
Theme:	buyer	buyer
Goal:	obtain-knowledgeable salesperson	obtain ad
Plan:	not-buy-ad	purchase-ad
Result:	achieve	achieve
Side+:		achieve-make-sale
Side-	block-make-sale	no result

Figure 6.4. Index for "Taxidermist"

For this index, the fear that the salesperson will be expected to be knowledgeable about the client's business could therefore be manifested by the student avoiding situations in which his or her specific business knowledge would be called upon. In other words, avoiding any technically-detailed discussion. The RCD is as follows:

WHEN the student is talking to a client in the pre-call stage,
LOOK FOR student to avoid technical discussion of client's business,
THEN TELL "Taxidermist"
AS a "Warn about pessimism" story, other-focused.

Warn about pessimism: Tell a story about a fear that was not realized when the student appears to have the same fear.

6.4. Warn about assumption

Lauren: Let's schedule the practice for Sunday. Dorothy is always around on weekends.

Kim: I'll call her just to make sure. Last time I scheduled something for the weekend, it turned out she was going to her in-law's.

Lauren has a plan with a certain assumption: she assumes that Dorothy will be available on the day of the practice. Kim recalls a story about a time when she made a similar assumption and it turned out not to hold. This is the **Warn about assumption** strategy. It is similar to the other perspective-oriented strategies in this chapter, except that instead of using outcomes that are desired or feared, it deals with the underlying assumptions that underlie decisions a student makes.

The category of assumptions covers all the kinds of expectations, explicit and implicit, that are not directly about the achievement or failure of goals. This is obviously a large subset of a student's state of knowledge, encompassing everything from background knowledge of business to social stereotypes and personal standards. Most of these attitudes would be very difficult for anyone to discern, particularly an observer such as SPIEL that can only see a student's actions. A student model, even a good one, would probably be of little help here, since students come from such a variety of backgrounds and experience that their assumptions in social situations would be very idiosyncratic.

To address a student's assumptions, the system has to attribute a set of mental states to the student. Where do these states come from? In a traditional intelligent tutoring system, they come from rules about the first principles of reasoning in a domain, including perhaps incorrect or buggy rules (vanLehn 1988). Such an approach has all of the difficulties of modeling that have been discussed.

Stories about failed assumptions provide a knowledge base about what particular assumption people have made and how they have failed. They allow, in effect, a case-based way of thinking about student modeling (Kolodner 1993). SPIEL looks for a student taking the same kinds of actions that faulty assumptions caused others to make. As in all of these strategies, SPIEL provides itself with an out by asking if it has diagnosed the assumption correctly.

See example 6.5. In this story, the salesperson assumes that there is nothing to learn from the client and that he can skip the step of trying to gather information, since similar attempts had yielded nothing in the past. SPIEL tells the story in a situation in which the student skips the pre-call step in a simulated sales call, because such an action betrays a similar assumption that the pre-call is not necessary.

6.4.1. Applying the student-focused version of the strategy

The index for this story is shown in figure□6.5. The anomaly in this index is in the negative side-effect slot. The salesperson assumes that there will not be any bad side-effects of skipping the pre-call step to save time with a predictable, non-buying client. What happens is that he fails to make a sale on a new portion of the account that he didn't know about.

Filter for Warn about assumption, student-focused: Select stories whose indices show student analogs making assumptions about their own actions that turn out not to hold.

- Student prepares an ad before pre-call.
- Student goes to see customer.

A warning about something you just did.

If you are assuming that you don't need to do a pre-call, you may be surprised. Here is a story in which an account executive made a similar assumption and was wrong.
I had a company once that was a property-management company. I had this company for several years. I had always pre-called the account and talked with the owner or the secretary trying to get new information trying to come up with ideas for the company with very little success. The company generally did not take my recommendations. I ended up going in and the secretary signed to keep everything as it was.
Through a few years of dealing with the customer in this way, I began to prejudge the account. One year I did not call the customer. I did not pre-call it. I didn't work up a program for the customer, because I felt, "Well, they're going to do what they've done for the past three years."
Then I went in to make the sales call. I heard the phone ring and they answered the phone, "This is the alarm company." My eyes just about popped out of my head. I had no idea that they'd gone into the alarm business. I had nothing worked up as far as artwork goes. I did not know who the primary person involved as far as taking care of that advertising would be. I felt pretty bad at that time that I shouldn't have prejudged the account.
What I did was I closed the property-management portion of the account. I tried to talk with the person that made decisions on the alarm portion. That person was busy and couldn't see me at that time, so unfortunately I had to leave with no sale. It was very close to the close of the directory and I had a lot of new customers that I had to see. I did try to call this customer back a couple of times but I never got them in. So I ended up missing the sale.
The assumption that a pre-call is unnecessary may be unrealistic.

Example 6.5. Telling "Not doing the pre-call" using the strategy **Warn about assumption, student-focused.**[8]

6.4.2. Inference implementation

The anomaly contrasts the absence of a negative side-effect, which was assumed, with the actual negative side-effect. The Actual column indicates that what is important is that the student failed to predict the negative side-effect of the plan on an important goal. What SPIEL must recognize is a similar failure of prediction on the student's part. The student stereotype again comes into play. Making sales is one of the most important goals that a salesperson has in the simulation. Saving time is not as important. If the student uses the "don't pre-call" plan, this is good evidence that either the student values the saving time goal more highly than the selling goal (unlikely) or that the student assumes that the selling goal will not be affected negatively by this plan. It has the following manifestation rule:

IF the evidence is needed that
the student predicts a positive impact on goal A as the outcome of a plan as opposed to a negative impact on goal B,
where goal B is more important than goal A,
A POSSIBLE MANIFESTATION is the execution of the plan by the student.

SPIEL must recognize that the student is using the plan of not using the pre-call step of the sales process. This is a case where building the evidence conditions forces SPIEL to modify the triggering conditions. The context of the story is the presentation, so SPIEL would normally look for evidence in the context of the preparation or pre-call stages. However, the plan explicitly calls for a modification of the standard steps of the process. Skipping the pre-call amounts to doing the preparation work at the time that the student should be doing the pre-

[8]Beckett, video clip #35.

Physical Setting:	presentation, service
Social Setting:	sales-target
Viewer:	salesperson
Perspective:	assumed
Agent:	salesperson
Anomaly Type:	side-

	Assumed	*Actual*
Theme:	seller	seller
Goal:	save-time	save-time
Plan:	don't-pre-call	don't-pre-call
Result:	serve	serve
Side+:		
Side-	no-negative-side-effect	block-make-sale

Figure 6.5. Index for "Not doing the pre-call"

call. What SPIEL looks for, then, is the student to skip from the initial stage into activities associated with preparation, namely the design of a proposal.

The RCD for this story/strategy combination is therefore

WHEN the student is in the initial stage,
LOOK FOR student to prepare an ad,
THEN TELL "Not doing the pre-call"
AS a "Warn about assumption" story, student-focused.

There are many cases when it will be more difficult to gather evidence about the student's assumptions. "Co-op success story," which was used to illustrate the **Demonstrate opportunities** strategy in the previous chapter, is a story with an anomaly that involves a failed assumption. The salesperson assumes that he will be able to capitalize on buying signs that the customer gave in the information-gathering phase of the sales process, but the customer turns out to be too fixated on the cost of the ad program for arguments about the benefits of advertising to be effective. The assumption here is that one type of argument will more persuasive than another. In most cases, the salesperson's beliefs about the relative merits of different kinds of arguments will simply not be discernible by the program.

This does not mean that SPIEL must give up on telling such a story. Even if the actual distinction in the index is too subtle for the system to gather evidence about, it is possible that an exaggerated version of the same distinction will be recognizable. An assumption that argument A will be more persuasive than argument B is subtle, but consider a more extreme version of this belief: argument A is convincing and argument B is worthless. A student's actions might betray such an extreme assumption. In the "Co-op success story," the exaggerated distinction is that the benefits argument will be all that is necessary. This can be recognized by the system as the student prepares his or her sales presentation. If the presentation omits material that would support arguments about cost-savings, even as a fall-back option, then it can be deduced that the student is assuming, as in the story, that the customer will be sold on the benefits.

SPIEL has a system of *exaggeration rules* to sharpen subtle distinctions into recognizable ones. The exaggeration rule here is

IF SPIEL CANNOT RECOGNIZE the student's belief about a certain value, THEN TRY TO RECOGNIZE the most extreme version of that belief.

With exaggeration rules, the system can try to make coarse, recognizable distinctions out of the subtle but difficult to recognize distinctions that may be present in indices. Even with such

- Student arrives for a pre-call appointment with a roofing contractor.
- The contractor's wife greets her and they carry on small talk .
- The student turns immediately to the contractor when he arrives.

A warning about something you just did

If you assume that Mrs. Swain will not have a role in the business, you may be surprised. Here is a story in which an account executive made similar assumption that did not hold:
I went to this auto glass place one time where I had the biggest surprise. I walked in; it was big, burly man; he talked about auto glass. So we were working on a display ad for him.
It was kind of a rinky-dink shop and there was a TV playing and a lady there watching the TV. It was a soap opera in the afternoon. I talked to the man a lot but yet the woman seemed to be listening, she was asking a couple of questions. She talked about the soap opera a little bit and about the weather.
It turns out that after he and I worked on the ad, he gave it to her to approve. It turns out that after I brought it back to approve, she approved the actual dollar amount. He was there to tell me about the business, but his wife was there to hand over the check.
So if I had ignored her or had not given her the time of day or the respect that she was deserved, I wouldn't have made that sale. It's important when you walk in, to really listen to everyone and to really pay attention to whatever is going on that you see.
An assumption that a spouse will not participate in the business may be unrealistic.

Example 6.6. Telling "Wife watching TV" using the strategy, **Warn about assumption, other-focused**.[9]

heuristics, it may still be possible that an index hinges on a belief that SPIEL does not have enough information from the simulation to recognize. For example, a story might be about how assumptions about the prevailing economic climate affect a customer. Because YELLO does not situate the student's sales call in a particular economic climate, the student's beliefs on this topic will come from outside the simulation context, and would therefore be very difficult to discern. In such a case, SPIEL would simply fail in its attempt to apply **Warn about assumption** and look for another storytelling strategy to use with the story.

6.4.3. Applying the other-focused version of the strategy

As with the other strategies here, **Warn about assumption** handles other-focused stories as well as student-focused ones. Consider the situation in example 6.6 (also seen in Chapter 2).

The student evidently assumes Mrs. Swain will not participate in the sale; otherwise, the time before Ed's arrival would have been used to better advantage. If the program has evidence that the student has a particular assumption about someone in the simulation, the **Warn about assumption** strategy calls for it to tell a story about a time when a similar assumption was wrong.

> **Filter for Warn about assumption, other-focused:** Select stories whose indices show the student analog making an assumption about someone else that turn out not to hold.

The index for this story is shown in figure 6.7. The anomaly in this index is in the theme slot. The salesperson assumes that the wife of the auto glass guy has a housewife-type role, not the role of a business partner, and so is surprised when she's the one who hands over the check.

6.4.4. Inference implementation

The marriage and partnership relations in the index setting are clearly shown to be important to the story's context by the presence of the housewife and business-partner themes in the index.

---

[9]McLean, video clip #305.

Physical Setting:	pre-call, service-business, small
Social Setting:	business-partner, seller, married-couple
Viewer:	salesperson
Perspective:	assumed
Agent:	wife
Anomaly Type:	theme

	Assumed	*Actual*
Theme:	housewife theme	business-partner theme
Goal:	be hospitable	contribute to decision
Plan:	make small-talk	evaluate presentation
Result:	serve	serve
Side+:		serve make sale
Side-		

Figure 6.6. Index for "Wife watching TV"

The recognition context looks for a pre-call situation that involves a married couple who are business partners, but this is a case where a little more generalization might be good.

Probably this story would be worth telling if the student ignores any member of the client's family in a family business. As we have seen, generalization quickly becomes problematic since SPIEL does not really know how general the lessons are that its stories teach. In order to do principled generalization, it would have to know more about the causality that would make the difference between a correct generalization and an overly-general one. It would also have to know what assumptions students are likely to make.

The assumption we are looking for is that the student assumes the spouse of the client will not have the role of business partner, which is a role critical for the formation of sales strategy, but instead to have the role of housewife, which is not important in sales strategy. How can this assumption be recognized? One possibility is that the student will specifically articulate an expected role for a person in the simulation. The student (if particularly sexist) might say something like: "Why don't we go into the next room Mr. Swain, so Lucy can get on with fixing dinner?" This would be a pretty good indication that the student doesn't believe she has a role that is critical to the sales process.

SPIEL has two manifestation rules to implement this idea:

> IF the expectation is that a certain person will have as a role that is not critical to a particular stage of the sales call, A POSSIBLE MANIFESTATION is an exclusion of that person from the main activity of that stage.
>
> IF looking for an exclusion of a person from an activity, A POSSIBLE MANIFESTATION is a statement of the desire to exclude that person from the activity.

In the pre-call context, the major activity is gathering of information and so a statement excluding Mrs. Swain from the information gathering conversation would satisfy this requirement. It is, of course, possible to exclude somebody without making a statement to that effect. In example□6.6, what the student does is simply to fail to engage her in informative conversation. There is a manifestation rule for this case also:

> IF looking for an exclusion of a person from an activity and that activity is the main activity for the context, A POSSIBLE MANIFESTATION is a failure to engage in the activity.

The RCD incorporates both of the possibilities discussed here:

WHEN the student is in the pre-call phase, and
the client has a business partner that is also a spouse, and
the client's spouse is present
LOOK FOR the student to exclude the spouse from the discussion, or
the student to fail to gather information from the spouse.
THEN TELL "Wife watching TV"
AS a "Warn about assumptions" story, other-focused.

Warn about assumption: Tell a story about an assumption that someone made which did not hold when the student appears to have a similar assumption.

6.5. Conclusion

The strategies in this chapter seek to use stories to address a student's possibly unrealistic expectations. This is an area where stories have great educational potential and where SPIEL's indexing scheme is particularly rich. However, in speaking to a student's expectations, these strategies are in the difficult position of making storytelling decisions based on suppositions about what the student is thinking. Since a detailed student model from which such suppositions could be drawn is out of the question, the strategies must go about gathering evidence in a somewhat oblique fashion. It must look for observable manifestations of students' thinking, analyzing the preparations that students make, such as designing prospective advertising.

These critiquing strategies sometimes must perform analysis to make the expectation identified in an index more clear. They use exaggeration rules to turn subtle distinctions between different expectations into gross differences that have obvious manifestations. They also search the index for points of contrast that may be easier to recognize than the main anomaly in the story, using anomaly derivation search.

7. Stories That Project Possible Results

7.1. Projection stories

Bert: What airline are you taking on your trip?

Kim: We're flying Continental.

Bert: Bad idea. Last time I flew Continental, we were delayed for two hours because of bad weather in Denver and they lost my luggage.

Kim describes a plan she is using and Bert tells his story to show her what might happen if she carries through with it. This is **Warn about plan**, a strategy that projects possible results. Such a story could also be used by the **Demonstrate risks** strategy if Kim went on the trip and reported having a good time. With this example we see the contrast between these two classes of storytelling strategies. The alternative-finding strategies step in when the student has done something and show how things could have been different. They say "You got lucky," or "You got unlucky." Stories that project possible results say instead "You're on the right track," or "Look out!"

Projection strategies try to help students learn what is one of the difficult skills in any complex domain: projection, envisioning what kind of outcome may result from a particular state. For example, a new salesperson might not realize that constant deferrals from the customer that put off the advertising decision probably mean that the customer has some serious objection that is not being addressed, an objection that could cost the sale. An experienced salesperson would anticipate that this may be the case and try to bring out the customer's objection. A story that shows the bad outcome that may lie ahead can stimulate students to think more carefully about their current direction.

Projection strategies, particularly **Warn about plan**, are probably the most familiar educational uses of stories. From the Old Testament to modern anti-drug propaganda, warnings are often couched as stories that describe the dire consequences visited upon those who have traveled the wrong path. This type of story is so common that there may be cause to doubt its effectiveness. The thesis behind the GuSS architecture is that students will learn best by actively discovering the consequences of their actions. Kim will learn about Continental if she takes her trip. A new salesperson who does not correctly read a customer's objections may learn from rejection. This is how learning-by-doing is supposed to work.

Still, there are a number of reasons why SPIEL needs to tell stories that project possible outcomes. The most obvious is that the outcome the story describes may be outside the scope of the simulated world. Suppose that a student uses a high-pressure sales tactic and gets the customer's reluctant agreement. Such a tactic makes the customer less likely to buy the next time around and in the long run, will result in lost business for the salesperson and the company. However, within the scope of the YELLO simulation, this counts as a success, since a sale was made. To make the long-term consequences vivid, SPIEL can tell a story about a salesperson who lost customers this way.

Projection strategies complement the simulation, teaching about circumstances beyond what the simulator can present to the student. The alternative-finding strategies also have this function because they can show stories with outcomes different from what the simulation produces, thereby getting the student to see some of the differences between the simulated world and the real experiences of others. Although the projection strategies don't contradict the simulation directly, they can show possibilities outside of its scope: letting the student get past the limitations of the simulation in another way.

Another reason to tell stories that project into the future is that the student may be unlikely to discover an error until a great deal of effort has been invested pursuing a course of action founded on a faulty premise. This work is not necessarily educational, even if it does make the impact of the eventual failure greater.

Suppose the student were making a car repair and had made a mistake early in the diagnosis process, diagnosing the wrong part as faulty. The student will discover that all of the work entering into making the repair is wasted only after the repair is completed and the problem remains. If an advisor could bring up a story that questions the diagnosis at an early stage in the process, the student has the opportunity to construct a revised approach before too much effort has been invested.

Using projection strategies places emphasis on getting the student to experience success. Such stories help the student to avoid traps, and to avoid misinterpreting what is happening in the simulated world. If novices are simply tossed into complex situations to sink or swim, many will fail and become very frustrated. Successful learning-by-doing requires that students be able to understand when they have failed and explain why. Novices may not always be able to do this, particularly when an error and the failure it causes are not obviously associated with each other.

Projection strategies therefore let learning-by-doing function where it works best: when there is a close association between actions and their outcomes. They step in to help when there is separation between what the student does and what results, a separation that a beginner may have difficulty bridging. The central issue for projection strategies is therefore this: how far away in time and effort is the possibility that the story projects? There are three possible answers:

Outcome is outside the bounds of simulation: If the outcome is such that it cannot arise within the simulated world, then SPIEL is always justified in telling a projection story.

Outcome far away: If the student would have to invest considerable time before the outcome would result, then it makes sense to tell the story ahead of time.

Outcome is near: If the possible outcome that the story is supposed to bring to the student's attention is likely to occur in the next few steps that the student takes, then the system looks for circumstances that would cause the outcome to be delayed. If such a delay occurs, the story can be told.

Part of the justification for SPIEL's storage and retrieval architecture is the independence of storytelling goals, the lack of interaction between tutorial opportunities to use stories. Because of the great similarity of the projection and alternative-finding strategies, this assumption is violated to a certain extent. There can be bad interactions between a strategy that uses a story to project a possible outcome and one that uses the same story to describe an alternative outcome. For example, if Kim goes on her trip and has a good time, it would be annoying (although perhaps, not unusual) for Bert to tell her his story again, with the **Demonstrate risks**-type strategy in mind. This is something a tutor should avoid.

One obvious prevention mechanism would simply be to prevent the storyteller from telling a story more than once based on the same index. This works in most cases. Imagine a story about a salesperson who feared the customer would be overly-fixated on price and used a tactic of not discussing the price until the end of the sales call, a tactic that eliminated the feared obstacle. The alternative-finding **Demonstrate opportunities** strategy might use this story if the student uses this delaying tactic and fails, in order to show that it is successful. The perspective-oriented **Warn about pessimism** strategy would use the story if the student shows evidence of fearing the customer's cost-consciousness. Suppose a student has this fear, hears the story, and then follows the story's advice, but fails to make the sale. The last thing such a student wants to hear is the storyteller with a reminder that someone used the tactic successfully. This would be perceived as insolence on the program's part. The student knows that an alternative outcome is possible because the story has already been told. Allowing only a single telling of a story prevents this type of blunder.

This simple bookkeeping scheme does not exactly resolve the interaction between the alternative-finding and the projection strategies. The "tell each story once" heuristic would always prefer the projection story since projection occurs in the midst of action before there is an outcome to compare against. So, strict adherence to the independence of storytelling strategies would prevent alternative-finding strategies from ever being used.

There is no obvious reason to prefer one of these strategies over the other, but since projection stories are so common, SPIEL tries to make sure that alternative-finding strategies have the chance to operate. It does so through an additional restriction on the projection strategies. If the student is certain to achieve an opposite outcome from that in the story, the story is not told even if the other storytelling conditions are met. The alternative-finding strategy is allowed to do its job. This certainty is achieved by seeing if the success or failure conditions for a plan have already been met.

7.2. Reinforce plan

Alex: What restaurant are you going to?

Sonya: Frontera Grill.

Alex: Great choice. Last time I was there I had their red snapper with pumpkin-seed molé. Fantastic.

In this example, Alex recalls a time when he used a plan similar to Sonya's and was very satisfied. The story reinforces her choice and, well in advance of any effort she would put into carrying out the plan, shows how the plan led to a good outcome in a particular case. This is the **Reinforce plan** storytelling strategy.

In a practice situation, students may not get positive feedback even when their plans are going well. For example, customers who become convinced by a salesperson's argument will not show their agreement for fear of losing some advantage in the sales situation. Stories about others' experience can show that the approach the student is taking has been successful in the past. The story thereby gives feedback when the simulation does not.

The **Reinforce plan** strategy does not have a counterpart in most intelligent tutoring systems, which are content not to intervene if the student appears to be on the right track. Tutors that strictly enforce a correct sequence of actions do not really need to give positive reinforcement since the absence of the tutor's intervention means that the student is doing the right thing. When this is not the case, tutoring system designers usually try to ensure that students get positive feedback in their practice, so reinforcing feedback from the tutor is unnecessary.

Of course, not everything the student does is worthy of a reinforcing story. It would be distracting (and probably patronizing) for a tutor to pat the student on the back after every action that could have a good outcome. However, in a complex domain like selling, many important variables, such as the customer's mental state, are hidden from the learner (Funke 1991). The student will often not see immediate cost or benefit from taking an action. For example, a salesperson making a sale to an inexpressive person may see the unresponsiveness of the customer as indicating a failure, when the customer is being persuaded by the sales presentation. A **Reinforce plan** story helps the student be confident that things are going well even when the appearance of the simulation doesn't necessarily support this belief.

In example 7.1, the student makes a sales presentation to the person who is managing the office, rather than to the actual decision-maker. The student will not know whether this

- The student is finding it difficult to get through to the lawyer in a law firm and instead makes an appointment with the office manager.
- In the presentation, she gives the office manager detailed facts and figures to support her recommendation.

A success story about a situation like yours.

Working with an office manager is sometimes important for making a sale. Here's a story about how an account executive succeeded using a similar approach.

This is a story about an account which I've called on for several years who is a large auto dealer in the Dayton marketplace. The owner of the business is a person who is very hard to get to. As a matter of fact, he goes by a code word even when his friends call in on the phone.

In this account, you meet with the general manager, always, to make the advertising decision. Now, you're never sure if general manager is making that decision or if he has to put it through to the code word man. Basically, this was an account where I have befriended and really become a confidant of the general manager. I realized that if a person goes to the extent of even having his friends call in under a different name to get to him as the owner, he's probably not going to see me as far as selling him Yellow Pages. I have become good friends with the general manager, talking about things like sports which he's very interested in and other subjects. He and I just hit it off.

It's got the point in our relationship where if something comes in Yellow Pages-wise, whether it be a mailer that says he should be in this book or somebody else selling him an ad, he refers everybody to me. It's come to the point where he has enough confidence in me that he will listen to whatever I want to tell him, whatever I think is the best thing he should do for his business.

Sometimes working with the office manager is the right thing to do. Keep working at it.

Example 7.1. Telling "Befriending the gatekeeper" using the **Reinforce plan** storytelling strategy.[1]

approach has been effective for quite awhile. The entire presentation must be made and then the manager and decision-maker will have to confer, after which the student will probably have to return on another sales call. SPIEL finds a story that shows a situation where a salesperson created a successful sales relationship with a person other than the decision-maker, and shows the student that it is possible to build a good business relationship even if the decision-maker is adamant about not being involved in sales presentations.

Another use of the **Reinforce plan** strategy can be demonstrated by the "$19/month is too much" story that was used to illustrate the **Warn about optimism** strategy in the previous chapter. There are many benefits of a good client relationship that do not appear in a few contacts with the customer. In selling Yellow Pages advertising, the biggest successes result from growing an account over many years as the customer becomes successful and attributes part of that success to the Yellow Pages. If the student makes a minor sale to a small business, he or she may feel that little has been accomplished. Telling "$19/month is too much" using the **Reinforce plan** strategy makes it clear that small incremental successes are very important contributors to long-term success. A **Reinforce plan** story can show students that their actions can have benefits that go beyond what is encompassed by the simulation. In this mode, it encourages students to think past the immediate goal of getting the sale, which is all a YELLO scenario captures, to issues like the long-term client relationship. Without the intervention of the storyteller, these issues would not come up.

---

[1]Beam, video clip #196

Physical setting:	presentation, large, retail
Social setting:	sales-target, employer
Viewer:	salesperson.
Perspective:	standard
Agent:	salesperson.
Anomaly Type:	plan

	Standard	*Actual*
Theme:	seller	seller
Goal:	make-sale	make-sale
Plan:	persuade client buyer	get gatekeeper help
Result:	achieve	achieve
Side+:		
Side-		

Figure 7.1. Index for "Befriending the gatekeeper"

Filter for Reinforce plan: Select stories whose indices show the student analog achieving a positive result.

Figure 7.1 shows the index for "Befriending the gatekeeper." The anomaly here is the contrast between the salesperson's standard, a preference to meet with the decision-maker, and what actually happened. The salesperson realized that someone who requires that his friends use a code name to get to talk to him was not going to give access to a salesperson, so the office manager's help became essential. The salesperson was successful in getting that help and making the sale.

The projection strategies are similar to the alternative-finding ones in their use of the index. They look specifically at the Actual column and try to gather evidence of similar events occurring in the simulation. Instead of trying to envision an outcome that is opposite, they ask the question: "In what situation would the student be executing a similar plan and be unlikely to discover the results without a lot more time and effort?" In other words, the strategy tries to predict the circumstances that would indicate a *remote outcome*.

As discussed earlier, if the student achieves a negative result instead of the positive prediction of the story, the **Demonstrate opportunities** strategy should tell the story to show that such failure need not always occur in this situation. To make sure this mode of storytelling is not always overridden by **Reinforce plan**, SPIEL checks that failure is not already a foregone conclusion. It sets up the RCD to look for the *failure conditions* of the plan. If the plan has already failed, it is best not to try to reinforce the student's use of it. Of course, the plan may fail after the story is told. In such a case, the "single telling" rule would prevent the **Demonstrate opportunities** strategy from presenting the story. It would be difficult to do better since predicting the outcome of an action in GuSS's open-ended simulation would require computing all possible outcomes of a given simulation state, something too time-consuming to do in the midst of the student's interaction.

Evidence conditions for Reinforce plan: Look for the student to have a similar goal and a similar plan, and the outcome to be far removed in time from the execution of the plan, and the plan not to be certain to fail.

7.2.1. Predicting indicators of remote outcomes

To tell "Befriending the gatekeeper," SPIEL needs to identify what would indicate that the outcome of the student trying to get the gatekeeper's help will be far removed. The result of getting the gatekeeper's help would be manifested by the buyer agreeing to an ad purchase on the basis of the gatekeeper's intervention. However, this agreement cannot take place in the same context as the execution of the plan, since the buyer is not there. The sales information will have to be communicated to the buyer in a separate context either by the gatekeeping employee or by the student in another visit, and then agreement may follow.

SPIEL uses *context comparison* rules to determine if two manifestations can occur in the same context. If two events occur in different contexts, SPIEL considers them sufficiently removed from each other that a projection story can be told. Here is the rule at work in this example:

> IF action1 is restricted by the absence of a person and action2 is restricted by the presence of that person, THEN action1 and action2 must occur in different contexts.

A presentation to the gatekeeper only counts as a presentation to the gatekeeper if the decision-maker is not present (otherwise it would be a presentation to the decision-maker with the gatekeeper looking on). The decision to buy must be taken by the decision-maker, therefore these two events must occur in different contexts. SPIEL is therefore justified in predicting that the student will not have any immediate feedback about the result of a presentation made to the gatekeeper.

SPIEL has to make sure that the plan has not already failed. If it has, then the story will be told by the **Demonstrate opportunities** strategy. To check for the possibility of such interference, the system uses *success* and *failure conditions* associated with a plan, the conditions under which the plan may be considered achieved or defeated. There are two failure conditions for the plan "get gatekeeper help." The plan will fail if the gatekeeper has the goal of preventing the goal that assistance is needed for or the plan can fail if the gatekeeper does not actually have any influence over the success of the goal. These are conditions that can be looked for within the mental state of the gatekeeper and the decision-maker. The final RCD therefore is:

> WHEN the student is in the presentation stage, and
> the student is talking to an employee of the decision-maker,
> LOOK FOR student to give a sales presentation to the employee, and
> the gatekeeper not to have the goal of preventing the sale, and
> the decision-maker to have the inclination to listen to the
> opinion of the gatekeeper regarding the sale.
> THEN TELL "Befriending the gatekeeper"
> AS a "Reinforce plan" story.

This RCD is very similar to the RCD seen for telling "Office manager only taking orders" using the **Warn about optimism** strategy. Anytime SPIEL tells "Befriending the gatekeeper" it will also tell "Office manager only taking orders." The student hears one story encouraging him to deal directly with the gatekeeper and another critiquing his hope that that person will in fact help. SPIEL present two widely-differing stories about the same issue. The student is thrust into a very real debate between experts,[2] and must decide independently which story is most directly relevant.

Reinforce plan: Tell a story about a successful plan to achieve a particular goal when the student has just started to execute a similar plan.

---

[2]It is interesting to note that both "Befriending the gatekeeper" and "Office manager only taking orders" were told by the same Ameritech Publishing account executive, Jim Beckett. His experience includes both positive and negative examples of relying on intermediaries in selling.

7.3. Warn about plan

The example at the beginning of this chapter showed **Warn about plan** in use. Bert describes his disastrous plane trip, a story that shows a negative outcome that resulted from flying Continental Airlines. This plan is similar to the one that Kim is using and so the story warns her that bad consequences have followed from other people using this plan.

The **Warn about plan** strategy is probably the most common storytelling strategy found in SPIEL. Stories told in this way pepper persuasive material of all types from religious tracts to "Reefer Madness." The message is "Look what terrible things happened to other people who did this. Turn back now while there's still time." It is also a time-honored tutoring strategy to interrupt the student when an incorrect action is taken. However, as we have discussed, SPIEL is not in the position of distinguishing correct from incorrect actions. It merely knows which plans have succeeded and which have failed in the stories in its story base, but this second-experience of others is important for a student without great experience of their own.

The **Warn about plan** strategy is a natural counterpart to **Reinforce plan**. That strategy attempted to show stories about positive outcomes when the student was not in a position to see evidence of such an outcome. **Reinforce plan** helps students avoid being discouraged when there is no obvious benefit to their actions. The same problem occurs for negative outcomes. The student may not be able to determine if something is wrong until disaster makes the problem obvious. **Warn about plan** brings up stories in advance of failure to show that others have run into trouble doing the same kinds of things.

Example 7.2 shows this strategy in use. As with **Reinforce plan**, the **Warn about plan** strategy is intended to provide feedback that is more immediate than would be possible if the student had to wait for an outcome from the simulation. In the example, if the student continues to press for the oversized ad proposal, the sale might eventually be lost altogether, but since the client has said he must refer to someone else, this could only happen after another sales call. The story shows a time when a similar approach on the part of a salesperson led to an immediate loss of the sale, suggesting that the student may be in more trouble than is immediately apparent.

> **Filter for Warn about plan:** Select stories whose indices show the student analog achieving a negative result.

As with **Reinforce plan**, there is a danger that this strategy could interfere with the student's natural learning process. A student who never saw the story would have an arresting expectation failure when the customer finally comes back with a rejection. As was true for the teller of "Would you buy this ad?", this could be a valuable learning experience. Since the **Warn about plan** strategy calls on the storyteller to intervene with a warning before bad things have happened in the simulation, the student (if he or she takes the storyteller's advice) will not have this experience first-hand, but rather second-hand through the story. The strategy therefore trades off dramatic but frustrating learning experiences against the more useful practical experience of setting a reasonable goal and working to achieve it.

> **Evidence conditions for Warn about plan:** Look for the student to have a similar goal and a similar plan, and the outcome to be far removed in time from the execution of the plan, and the plan not to be certain to succeed.

- The student does a minimal amount of research before talking to a roofing contractor client.
- The student doesn't gather very much information at the pre-call stage.
- Back at the office, the student prepares two ads, one that is much larger than the contractor's current ad and a second large ad for a different Yellow Pages book, a very large ad campaign.
- When the student presents these to the client, the client says "You know, I really have to talk to my dad about this..." and is inwardly very doubtful of the value of such a large expenditure.

A story about a failure in a situation similar to yours.

You made a recommendation that was much larger than the client's expectations. Here is a story in which doing that led to problems.

I remember my first year 1970. I was on my first Yellow Page sales canvass. In those days, they didn't give you a lot of time to show ability and I wasn't doing very well. My manager told me that I had one week to start producing or they were going to let me go.

I called on a graphic artist in Indianapolis. He had a 2HS, a one-inch ad. I walked in, asked two minor questions, and I laid down a quarter-page piece of spec in front of this man and told him he needed this ad. The man looked at me and he said, "Would you buy this ad?" He turned it right back to me. I didn't know what to say. It shocked me. Finally, I said, "No, I wouldn't." He didn't need a quarter page; a one-inch ad is what he needed. The gentleman proceeded to give it to me, up one side and down the other. He told me that I was there for my own greedy interests, trying to make commissions instead of taking care of his advertising and caring about him. He said, "If you're ever going to make it in this business, you'd better start paying attention to your customers."

I walked out of that call a different salesman because I realized then that the only way to sell Yellow Pages is to sell what the customer needs, not what I need. Learning that lesson turned my sales career around. I started making sales, and by the end of the canvass, I was one of the top producers.

You made a recommendation that is much larger than the client's expectations. That might not be a good idea.

Example 7.2. Telling "Would you buy this ad?" using the **Warn about plan** storytelling strategy[3]

Of course, as was the case with the previous strategy, it is also possible that the failure referred to in a story is not something that could occur within the bounds of the simulation. The student would never encounter such an outcome and never have the associated learning experience. In such a case, SPIEL is always justified in telling a **Warn about plan** story.

SPIEL has to determine "how far away" the negative outcome might be. Like **Reinforce plan**, the strategy first reasons about the manifestations of the plan and result, and does context comparison. (See the index for "Would you buy this ad?" in figure 7.2.) In this story, the failure of the sales-related goal of making money would occur in the same context as the presentation. This outcome is not far away and does not lend itself to the **Warn about plan** strategy.

There are cases in which the decision on a proposal, normally occurring right away, would be delayed. If this happens, the outcome could indeed appear only after significant additional effort by the student. What **Warn about plan** must look for is an indication of delay. The customer's deferral is an example of this kind of indicator. By putting up a smoke-screen, the customer has greatly lengthened the amount of time it will take the student to confront the real objection.

---

[3]Gant, video clip #149.

Physical setting:	presentation, professional, small
Social setting:	sales-target
Viewer:	salesperson
Perspective:	wanted
Agent:	salesperson
Anomaly Type:	result

	Wanted	*Actual*
Theme:	seller	seller
Goal:	maximize monetary gain	maximize monetary gain
Plan:	propose-big-ad	propose-big-ad
Result:	achieve	block
Side+:		
Side-		block obtain rapport

Figure 7.2. Index for "Would you buy this ad?"

SPIEL has *outcome delay* rules that make this kind of inference. Given a plan and an outcome, the rule gives indicators which would show that the outcome, while normally proximate to the plan, will be more remote. Here is the rule that is used for this story:

IF plan is a presentation plan, result is a sales outcome and the target is the decision-maker, THEN A DELAY IS LIKELY IF the client defers the decision to another time or another person.

It must also determine that the plan is not certain to succeed (in which case **Demonstrate risks** would tell the story later). The success conditions for the presentation of a proposal are that the decision-maker believes the value and advisability of the proposal. The RCD therefore is:

WHEN the student is in the presentation stage, and
talking to the decision-maker,
LOOK FOR the student to propose a large ad, and
the client to defer the decision, and
the client not to believe the value of the proposal, and
the client not to believe the advisability of the proposal.
THEN TELL "Would you buy this ad?" AS a "Warn about plan" story.

Warn about plan: Tell a story about an unsuccessful plan when the student has begun executing a similar plan.

7.4. Conclusion

The two strategies in this chapter tell stories that project possible results of student's actions. They are, in a way, the most familiar types of stories to be found in teaching situations: essentially positive and negative exemplars of what the student is setting out to do. These strategies must do their work without being able to predict exactly what is going to happen to the student: they cannot envision all possible simulation outcomes. Instead, the projection strategies use several kinds of knowledge about the simulation. They use simulation scope rules to determine whether a projected result is beyond what is represented in the simulation. In such a case, the story is always good to tell. For cases that are less straightforward, SPIEL must try to infer the length of time it would take for the student to encounter the result the story projects. It does not make sense to tell a story to make an indirect point if the simulation itself is about to give the student the same lesson in a very direct way.

To look for distance between events, SPIEL has knowledge of contexts in which events can occur and the temporal relationship between contexts. An outcome that occurs in a different context than the one the student is currently in is considered far enough away that a projection story is useful. A projection story is still useful even if the outcome would normally occur in the same context, because circumstances may cause the outcome to be delayed or hidden from the student. SPIEL uses outcome delay rules to infer actions that indicate an outcome will be delayed to justify the telling of a projection story about it.

Projection strategies are an exception to the rule that storytelling opportunities do not interfere with each other. Although SPIEL avoids telling the same story twice, this heuristic alone would result in unsatisfactory performance. An opportunity to tell a story told using a projection strategy will always precede an opportunity to use an alternative-finding strategy, and alternative-finding strategies would never have a chance to operate. SPIEL tries to prevent this interference when it can by preventing the operation of projection strategies at those times when it is clear that an alternative-finding strategy will tell the same story.

8. Stories That Explain Others' Behavior

8.1. Other-directed storytelling strategies

Mary: Why does Max keep bugging me to go out with him? You'd think after three weeks he'd get the idea.

Dolores: I remember that I went to the movies with him once and he thought it meant we were an item. It took months before we could have a reasonable interaction.

When problems arise in social interactions, often what must be explained is the behavior of the other people involved; knowing what others are doing is just as important as knowing what to do oneself. In this example, Mary has trouble understanding why Max is still hounding her after she's given all of the appropriate "not interested" signs. Dolores's story, while it doesn't really explain what the problem is, confirms Mary's observations through reference to a similar situation. The message is that Max is just one of those guys who can't take a hint.

Students need to hear stories about those with whom they are learning to interact. Orr (1986) found that, although their field is a technical one, copier technicians have a wealth of stories about the people they interact with in the course of their duties. In selling, mistakes are often caused by a failure to understand the client's point of view. The bizarre and surprising expectations and actions of clients are an important part of the lore of salespeople. In the case of Yellow Pages salespeople, most of the clients are small-business people. Their business concerns and their attitudes with respect to advertising may not have been frequently encountered by students in their daily lives, but student must become attuned to them in order to build rapport with customers and to construct sales arguments that speak to clients' needs.

The storytelling episode in the example is triggered by an explicit question from Mary asking for clarification of Max's behavior. SPIEL does not get such an explicit indication of the student's reaction. SPIEL has to infer that the student may not understand some character's action in the simulation, since the only reason to explain something is if the student is not likely to know about the phenomenon being explained. A student who is personally acquainted with many lawyers might not be surprised to hear a story about an attorney who was unfriendly towards salespeople, but someone with few dealings with the profession might need the explanation the story provides.

Other-directed stories only make sense if they are about things outside of normal experience, yet every student will have their own baseline of normal behavior. The answer is obviously not to try to model each student's experience, given the complexity of an individual's social experience and knowledge. As with perspective-oriented strategies, SPIEL uses a simple solution to this complex problem. It maintains a stereotype representing the average incoming student's stereotypes about those with whom they are interacting. It will only tell a story if it believes that the character's actions fall outside the normal range of what the student would likely expect. For example, a story about a customer who is concerned about the cost of advertising would not be exceptional, but one who wanted the salesperson to become a personal friend would be.

SPIEL has a static *student knowledge stereotype* that represents what students who are new to a domain of expertise can be expected to believe. These expectations are organized using a set of stereotypical roles in the domain being studied. In YELLO, these are the decision-maker, gatekeeper, competing salesperson, etc. For each role, there is a set of standard goals, a standard set of plans for achieving these goals within particular contexts, and a set of standard beliefs. This knowledge is used by the filters that choose which stories can be told using the other-directed strategies. A story in which someone violates the stereotype associated with their role is assumed to be of interest to a student.

Other-directed strategies differ from all the others in that they largely address what characters other than the student are thinking. To achieve this, they must break the guideline discussed in Chapter 4 that the other strategies tend to follow, namely the reliance solely on externally-observable states of the simulation. The reason that other strategies can rely only on what the student can observe is that students can employ their own understanding of the situation in the simulated world to interpret the analogy drawn by a storytelling strategy when a story is told. Students will reject stories whose advice runs counter to their understanding of what is happening. When the analogy is drawn to someone other than the student, there is no easy way for the student to reject the analogy.

Since what the storyteller alludes to in its storytelling cannot be independently verified, students will tend to trust the tutor, believing it would not knowingly lead them astray. It is important that SPIEL live up to this expectation as best it can. Therefore the other-directed strategies always verify the conditions that they report by inspecting the internal representations of the characters in the simulation. The tradeoff is that the other-directed strategies may occasionally give students "inside information" about the workings of the scenario they are in, but this is better than giving bad advice.

In the other strategies discussed so far, the major portion of the effort of the strategy is directed towards looking for recognizable manifestations of the student's thinking. This problem does not arise in the same way for other-directed strategies, since the characters' mental states can be directly inspected. Just looking at internal states, however, does not fulfill the mandate of the other-directed strategies to explain things that a student may not understand. A student has to see a character act on the basis of a plan or belief before there can be a failure to understand. SPIEL therefore looks for visible manifestations of mental states in addition to inspecting characters' representations.

8.2. Explain other's plan

Rhonda: Why isn't Max here yet?

Bert: Well, last time I invited him to come out to hear some music, he spent a couple hours on the phone trying to get somebody to come with him. He feels inadequate if he doesn't have a date.

In this example, Rhonda is having trouble understanding why Max hasn't joined them at the appointed time. Bert tells a story by way of explanation, describing a plan that Max used on a previous occasion that caused him to be late. The story accounts for Max's lateness in terms of his insecurity about being without a date. This is the **Explain other's plan** storytelling strategy, which calls on SPIEL to bring up a story when it can help explain to the student the actions of another.

As I have discussed, students will often have trouble understanding what characters in the simulated world are up to. A character may have a goal that the student did not anticipate. Imagine, for example, a student who is trying to get access to a decision-maker and who encounters a manager who demands to be involved in the sales process. The most common explanation would be that the person is a "control freak" who wants to be in on everything. An alternative possibility that SPIEL can raise by telling a story is that the person is a frustrated artist who wants to have creative input on the design of the ad. This is a goal that a novice salesperson might never expect someone to have.

It may also be that a character is using some unexpected means of achieving a well-known goal. This is what happens in example 8.1. The client is being abusive because he thinks that doing so gives him an advantage over the salesperson and may result in a better deal.

- The student starts a pre-call with an attorney who is extremely rude and dismissive.

Do you understand what [the attorney] is doing?

Sometimes an attorney will be rude to salespeople as a matter of policy. Here is a story about a time when that took place:

I cringe at the thought of this one attorney I did deal with. It was basically when I had first come into sales. It was my breaking in point there. I had talked to an attorney who had been in the business for 35 years. His ego was as big as... He intimidated me very much. Being new, I didn't know how to handle it at first. He virtually tore me down to bits. I was ready to tell him "Good-bye" and that's it. Hang up on him.

After awhile talking to him, he explained to me that he does this to all his representatives. He felt that his 35 years in the business gave him the right to knock people down. I told him, I said "No, you don't have no right to do this to me. All I did was call. I wanted to improve your program for you, and be a friend." He was that bitter. It was his ego, basically. He was bitter about people and he wanted to let them know that he had earned the right to do this to people. At the end of it, he did apologize to me. He gets carried away, he told me. He told me he was sorry for what he did, but it's in his nature to do this.

He sent me his book. He was a published author. We got to talking (this was on the second call). Because after the first call, I just gave up on him for awhile. I called him back the second time. We did talk. He let me know more about himself, and what he had in mind for his Yellow Page advertising program. We put together a nice display ad for him. The following years I did have him back again. In fact, I had him for three years after that. We got to be real good friends. From that point we grew his advertising program and we did turn out to be good friends.

[The attorney] may be rude to salespeople as a matter of policy in order to gain an advantage. If you don't know, you should try to find out.

Example 8.1. Telling "Obnoxious attorney" using the **Explain other's plan** storytelling strategy.[1]

To determine if a story can be told using this strategy, SPIEL uses the student knowledge stereotype. The agent in the index is the person who performed the actions that the story describes. SPIEL extracts the roles that the agent played in the sales process in the story, and looks at the stereotypes associated with these roles. The standard goals and plans that are associated with the agent's roles are compared against the goal and plan in the Actual part of the index. If a story contains someone using a typical plan to achieve a typical goal, it will not be considered worth telling, since the student will probably be familiar with what the story shows. Only if the story contains goals, plans or combinations of them that violate the stereotype is it useful for the **Explain other's plan** strategy.

For example, the student knowledge stereotype contains the standard expectation that people will only use abusive language if they're angry and they want you to know it. In the story, the plan of abusing the salesperson is not serving a goal of expressing anger; the goal is to gain an advantage in the interaction. The plan is therefore not serving the goal that the stereotype associates with it, which is considered a violation of the stereotype and an indication that the story is worth telling using **Explain other's plan**.

Filter for Explain other's plan: Select stories whose indices show someone who is not the student analog with a goal and plan that are not compatible with the student stereotype for that role.

SPIEL needs to gather evidence that a character in the simulation has the goal and is using the plan suggested by the story. This much is made fairly simple by the fact that **Explain**

[1]From interview with Paul Cuglowski, video clip #243.

Physical setting:	pre-call, professional, medium-sized business
Social setting:	sales-target
Viewer:	salesperson
Perspective:	standard
Agent:	attorney
Anomaly Type:	plan

	Standard	*Actual*
Theme:	buyer	buyer
Goal:	preserve-civility	gain-advantage
Plan:	be-polite	be-verbally-abusive
Result:	achieve	achieve
Side+:		
Side-		block-personal-comfort

Figure 8.1. Index for "Obnoxious attorney"

other's plan is allowed to check characters' internal states. In addition to verifying a character's intentions, SPIEL must also be sure that the student sees the character acting on these intentions. Otherwise, what does the story serve to explain? The story will not be relevant unless the student sees someone in the simulation do something unusual that is explained by the story. Evidence conditions for **Explain other's plan** therefore look for a character's hidden intentions and for external manifestations to make sure that the student has seen evidence of those intentions.

> **Evidence conditions for Explain other's plan:** Look for a character in the simulation with a role similar to the agent in the story to have a similar goal and plan to those in the actual part of the index and to act on the plan while the student is observing.

SPIEL looks for a character to have internal states that represent the same goal and plan as in the index. (See the index for "Obnoxious attorney" in figure 8.1.) The goal here is to gain social advantage over the salesperson and the plan is the employment of verbal abuse. SPIEL uses manifestation knowledge to look for actions that would manifest the plan, with the additional restriction that the student must observe those actions. SPIEL knows that certain plans can be directly observed in action, and do not require special reasoning about their manifestations. The plan of using verbal abuse is such an *observable plan,* so it can appear in the RCD.

WHEN the student is in the pre-call stage, and
    the student is talking to the decision-maker
    LOOK FOR the decision-maker to have the goal of achieving social
            advantage over the student, and
        the decision-maker to have the plan of using verbal abuse, and
        the decision-maker to be verbally abusive
    THEN TELL "Obnoxious attorney"
        AS an "Explain other's plan" story.

> **Explain other's plan:** Tell a story about a plan that the student might not know about when the student has just observed some agent execute a similar plan.

8.3. Explain other's perspective

The example that opened this chapter showed Dolores using the **Explain other's perspective** strategy. When Mary had trouble understanding Max's behavior, Dolores told a story about a faulty expectation Max once had about her. Because the actions are similar, the implication is that Max may have a similar delusion about Mary.

This strategy uses the contrast between expectations and reality that the perspective-oriented strategies in Chapter 6 used, but with a different effect. The perspective-oriented strategies try to help the student avoid unrealistic expectations by telling stories that show how similar expectations have failed in the past. **Explain other's perspective** tells stories about other people's unrealistic expectations, not to show the error of such expectations, but to explain to the student what may lie behind the actions the student observes.

This is particularly important with respect to people's expectations of the student's role. Being new to the role, a new salesperson may not realize, for example, that a client who is talkative and particularly friendly may have an expectation of a social relationship that extends beyond the professional. The system has stories about salespeople who angered clients by failing to understand and manage such unrealistic expectations. These stories can help the student see what may lie behind the customer's reaction and what it may mean for the sales relationship.

Chapter 6 showed three different storytelling strategies, each of which addressed a different kind of perspective on the part of the student. Yet, there is only a single perspective-oriented strategy having to do with others. There are two reasons for this.

Impact on the student: The strategies that warn students about their own expectations all have different educational impacts on the student. If the expectation is a fear, the strategy serves to bolster the student; if a hope, it serves to dampen aspirations. When addressing stories to the perspective of others, there is really only a single kind of impact: to show the student the viewpoint of this other person.

Type of manifestation: The perspective-oriented strategies in Chapter 5 differed in the kind of manifestations they used to indirectly detect different types of expectations: optimism was recognized when students reached too high; fear, when they shrank back. **Explain other's perspective** has the advantage of being able to inspect characters' mental states and so can detect their intentions in a uniform and simple manner.

In example 8.2, the student encounters a customer who is adamant about removing her advertising completely. SPIEL tells a story about a customer who had a similar reaction because he was upset by the service he got from previous sales representatives and expected that this year's salesperson would be no better. The student is then encouraged to try to find out if the customer has a similar fear of being taken in.

Like the perspective-oriented strategies, this one focuses on the anomalous expectation in the index and looks for evidence that a character in the simulation may have that expectation, but it must also be sure that the expectation is something that the student will find anomalous. **Explain other's perspective** has to verify that the story violates the student knowledge stereotype. SPIEL looks at the stereotype associated with the role played by the viewer, the person in the index who had the anomalous expectation. If the expectation in the story violates one of the beliefs associated with the stereotype of that role, then the story is a candidate for telling with this strategy.

Physical setting:	retail, pre-call
Social setting:	sales target
Viewer:	buyer
Perspective:	feared
Agent:	salesperson
Anomaly Type:	goal

	Feared	*Actual*
Theme:	seller	seller
Goal:	make sale worthless	make sale best for client
Plan:	persuasion plan	propose small ad
Result:	serve	serve
Side+:		
Side-	threaten get value for money	

Figure 8.2. Index for "Badly-served customer"

> **Evidence conditions for Explain other's perspective:** Look for a character in the simulation with a role similar to the viewer to have an expectation similar to the anomaly and to act on that expectation while the student is observing.

This storytelling strategy, like the perspective-oriented ones in Chapter 6, tries to detect an expectation similar to the anomaly in the index. For "Badly-served customer," the contrast is between the goal the customer feared the salesperson would have and the one he actually had. There are two basic areas of difference between the feared and the actual goal. One is the degree of conflict between the salesperson's goal and customer's goal of having advertising that meets his needs: the fear is that there will be great conflict. The other aspect of the fear is the value of the resource (advertising) that the salesperson is offering: the fear is that the resource has no value.

The RCD must also test for manifestations of these fears, since the student must see evidence of the expectation before a story will be a sensible intervention. SPIEL has the following two manifestation rules that operate here:

IF the issue to be recognized is a belief in goal conflict and the other's goal is a social goal,
    A POSSIBLE MANIFESTATION is a refusal to assist in other's goal.
IF the issue to be recognized is a belief in the value of a resource,
    A POSSIBLE MANIFESTATION is a negative response to a discussion of the value of the resource.

The RCD therefore looks like this:

WHEN the student is in the pre-call stage, and
    the student is talking to the decision-maker,
    LOOK FOR the client to have the expectation that the salesperson is
            in conflict with their interest, and
        the client to refuse to co-operate in the sales process, and
        the client to make negative remarks about the value of Yellow
            Pages advertising.
    THEN TELL "Badly-served customer"
    AS an "Explain other's perspective" story

> **Explain other's perspective:** Tell a story about an expectation that the student might not understand when the student has just observed some agent act on the basis of a similar expectation.

8.4. Stories about others

The other-directed strategies complete the set of SPIEL's storytelling strategies. They address themselves to the experiences of people in roles different from the role played by the student, such as the customer in YELLO. They compare the real-life "others" that the student will encounter in the world with the characters in the simulated world. They differ from all of the strategies described in preceding chapters that draw analogies primarily between the main agent of a story (as identified in the index) and the student.

There are only two of these strategies compared with the many that address the student's role, but they are crucial to a student's education in an area of social expertise. Students need to have an appreciation for the circumstances of others, and they need to understand how they will be perceived by others in the role they are learning to play.

Students will inevitably transfer beliefs and expectations about people from their everyday social experience with social situations to the new situations they encounter. They must learn how much of this knowledge they can carry over, how much they must discard, and what new knowledge they need. The intent of the other-directed strategies is to point out the gaps in students' untutored expectations and show actual examples from which new generalizations can be drawn.

To use these strategies, SPIEL has to represent the preconceived beliefs of the student. This is the role of the student knowledge stereotype. The student stereotype lists the common roles that people play in the domain being studied and associates with them goals, plans and expectations. The experience that practitioners report in their stories is often at odds with the preconceptions of newcomers. SPIEL assumes that stories that do not report such conflict are probably not interesting to the average student, presenting only those that violate the stereotype.

There are some significant problems with the stereotype model. Every student has his or her own expectations about the motivations of others, based on personal experience. Some students may understand perfectly well actions that run contrary to the stereotype and have difficulty understanding beliefs that the stereotype takes for granted. This, of course, is the reasoning behind student modeling that is dynamic and tailored to each student (vanLehn 1988).

Student modeling as normally practiced, however, does not give much leverage of the problem of idiosyncratic student beliefs. In Burton's DEBUGGY (1982), for example, the system has a set of "bugs," rules for incorrect behaviors that students frequently exhibit. This is possible because the task of addition is sufficiently small that there are only a limited number of ways to do it wrong. In a social domain, students' expectations are part of their personal systems of belief about others. For a model to identify with any precision what actions the student sees as anomalous, it would have to understand each student's own belief system. Even if such a model could be constructed in principle, it would require much more time than can be devoted to the task within an educational simulation.

A slightly more flexible version of this strategy could be built if the system had a library of student knowledge stereotypes, each based on a different class of student belief system, not unlike Rich's user stereotypes (Rich 1986). There might be one for students who believe that others understand their job in the same way they do, and a different stereotype for students who have a more relativistic outlook. The other-directed strategies could then create sets of tutorial opportunities corresponding to each stereotype. At retrieval time, the system could ask the student to perform a diagnostic test to find out the stereotype that matches best, and use only the tutorial opportunities generated using that stereotype.

Table 8.1. SPIEL's storytelling strategies

What the strategy does:	Storytelling strategy	Story is about	Tell story when
Shows alternatives.	Demonstrate risks	the negative result of a particular course of action.	the student has executed a similar course of action with good result.
	Demonstrate opportunities	the positive result of a particular course of action.	the student has executed a similar course of action with poor result.
	Demonstrate alternate plan	a successful plan to achieve a particular goal.	the student executed a very different plan and failed to achieve goal.
Critiques student's perspective.	Warn about optimism	a desire that someone had that was not realized	the student appears to have the same hopes.
(Note: each of these strategies has two types: student-directed and other-directed.)	Warn about pessimism	a fear that someone had that did not materialize.	the student appears to have the same fear.
	Warn about assumption	an assumption that someone made that did not hold.	the student appears to be making the same assumption.
Projects possible results.	Warn about plan	an unsuccessful plan.	the student is executing a similar plan, but won't see the result soon.
	Reinforce plan	a successful plan.	the student is executing a similar plan, but won't see the result soon.
Explains others' behavior.	Explain other's plan	a plan that the student might not know about.	student has just observed someone execute a similar plan.
	Explain other's perspective	a belief that the student might not know about.	student has just observed someone act on the basis of a similar belief.

8.5. Choosing among storytelling strategies

The survey of SPIEL's storytelling strategies is now complete. Table 8.1 summarizes them. With these strategies the system can examine the indices that label tutorial stories, determine how these stories ought to be presented, and generate descriptions of the conditions under which they should be retrieved.

One issue that has been conspicuously absent in this discussion is the question of strategy choice: when it is appropriate to employ what strategy? This omission is deliberate. The basic philosophy behind SPIEL's storytelling intervention is that any relevant story is worth telling, and therefore any storytelling opportunity should be capitalized on. Strategies are simply means of getting stories across. SPIEL does not set itself up as an arbiter of right and wrong in selling. It attempts to connect the student's experience with that of others using the storytelling strategies.

This architecture is a response to the nature of the tasks for which SPIEL has been employed. Expert social tasks such as selling are "weak-theory" domains (Porter, Bareiss and Holte 1990), where there is no complete causal model of the phenomena being studied. Worse yet, from the standpoint of a computer tutor, these skills are highly dependent on pre-existing knowledge of social norms and practices, and have a significant cultural component. It is often impossible, even for a human expert, to say that a certain social action is definitely wrong since there are so many ways that social goals can be accomplished. For example, is it a good idea to talk with Mrs. Swain about fishing? No, since she's not that interested in fishing. Yes, because it serves as an ice-breaker for getting conversation going.

SPIEL has, in effect, a case-based model of the task that the student is learning. It does not know what is right or wrong, only what others have reported doing. It operates in this way because I chose to build a system to come at the problem of tutoring from a completely opposite approach from that used in intelligent tutoring systems. Because such tutors have always needed to have a model of the task the student is learning, they have never been applied to complex social tasks, where the construction of complete expert models has proved intractable.

This case-based approach does mean that what SPIEL teaches will be very dependent upon its story base. Suppose SPIEL has a story about a salesperson who flirted with a customer and was successful in making a sale. If the student does a similar thing and gets into trouble, the **Demonstrate opportunities** strategy would call for system to tell the story as reassurance that flirting sometimes works. However, flirting with customers is probably something that a salesperson should avoid, irrespective of a particular success story. Currently, the only way to avoid this problem would be to leave the flirting story out of the case base.[3] Because the system is so uncritical of what it tells, only experienced and successful salespeople contributed stories to YELLO's story base.

It would be possible to build a storyteller that used storytelling strategies more discriminately. A system that had some heuristics about right and wrong could search for storytelling opportunities selectively. For example, if a student appears to be making serious mistakes, the storyteller might look only for opportunities to tell stories that make a negative point: opportunities to use the various "warning" strategies, not the positive strategies. In the case of the student who is flirting with a customer, a tutor could use such a heuristic to suppress this use of **Demonstrate opportunities**, and prevent a flirting success story from being told.

Steps in this direction must be made with caution. If there were really good heuristics about how to sell, people would be able to teach it much better than they do (Craig and Kelley 1990). The "rules of thumb" that experts give are frequently ignored under the appropriate circumstances. For example, salespeople often say that in selling one should always "aim for the sky," try to sell as much as possible. If pressed, they will admit that with certain customers and certain accounts, this is a bad idea. Blindly building in simple heuristics may cause useful remindings to be unnecessarily rejected.

Instead of developing heuristics for the suppression of storytelling strategies, a better approach might be a balancing heuristic. The system could label stories that contain questionable or extremely atypical events, and present them only if a balancing story or stories can be presented at the same time. To avoid giving the impression that flirting is a good tactic, SPIEL could tell the flirting success story accompanied by a flirting disaster story. A balancing heuristic would make sure that relevant stories are not hidden from the student while not allowing unusual experiences to unduly bias the student.

8.6. What's next

When a library of stories has been processed using storytelling strategies, SPIEL knows when and how the stories should be presented. This is the hard work of implementing the case-based teaching architecture. Chapter 9 completes the picture of SPIEL's storage and retrieval

---

[3](Lesgold and Lajoie 1991) makes the point that anecdotal reports may be too persuasive for their own good, leading practitioners to ignore other kinds of evidence.

phases, describing the two remaining components: rule generation, the task of turning the recognition condition descriptions created by strategies into recognition rules, and rule application, the task of using these rules to perform retrieval and presentation.

9. Rule-based Recognition of Storytelling Opportunities

9.1. An overview of rule generation

As characterizations of storytelling opportunities, recognition condition descriptions (RCDs) tell SPIEL what to look for. The system must go beyond these descriptions to be able to recognize opportunities. Knowing what to look for is not the same as knowing how to look. Consider looking for an opportunity to help a friend. A description of what to look for is simple: a situation in which one can take action that contributes to the achievement of one of the friend's goals, but it is not a trivial matter to be able to recognize all such situations. Opportunities can appear in many novel and subtle guises (Birnbaum 1986).

Although SPIEL can characterize storytelling opportunities, it still must be able to recognize them. However, it does not have to deal with novelty. The environment in which it operates, the GuSS simulation, is a very constrained subset of the social environment. RCDs contain general descriptions of situations, such as "the student ignoring a person's input." In the real world, there is an indefinite number of ways that such an event could occur: it would appear one way in a conversation during a sales call, another way in an act of letter-writing, still another way while having dinner with the client. The simulation limits the number and type of manifestations of a storytelling opportunity; only a few possibilities will be present within any GuSS application. For example, in YELLO, students interact with clients at their places of business or over the phone. They do not write letters. Therefore, a procedure for recognizing that the student is ignoring someone in YELLO can safely ignore whole classes of possibilities such as those that involve letter-writing.

SPIEL capitalizes on the constraints inherent in the environment to build simple rule-based recognition procedures for storytelling opportunities. This chapter describes how SPIEL performs the conversion of opportunity descriptions into recognition procedures, turning RCDs into GuSS rules, and how these rule-based recognition procedures are invoked to retrieve stories.

9.1.1. Rule generation as design

Rule generation in SPIEL can be thought of as an automated design process. The process begins with a general idea of what is desired, and this general design is refined in stages as additional constraints are added until it is completely specified. The constraints on the recognition procedure come in the form of more and more specific knowledge about the form of the opportunity to be recognized and the structure of the environment in which it may appear.

Figure 9.1 shows the steps in this design process and the knowledge that informs each. I have included for completeness the strategy application process covered in the previous chapters. Strategy application takes a description of a story in terms of its index and produces a description of an opportunity to tell that story. In doing so, it uses three kinds of knowledge: (1) knowledge about how a story can make an educational point, the storytelling strategies; (2) knowledge of how beliefs and intentions may appear in action, general manifestation knowledge; and (3) a set of default assumptions about the student, the student stereotype.

Process	Input	Output	Knowledge type
Strategy application	Index	RCD	Storytelling strategies General manifestation knowledge Student stereotype
RCD elaboration	RCD	expanded RCD	Application-specific manifestation knowledge
Rule spec. creation	eRCD	Rule spec.	GuSS-specific rule application knowledge
Rule creation	Rule spec.	Rule set	Application-specific translation knowledge
Rule optimization	Rule sets	Optimized rules	Rule comparison and combination knowledge

Figure 9.1. Processes and knowledge use in SPIEL

The second process is that of RCD elaboration. Here the system takes into account the task the student will be performing and how that task is realized in the learning environment. These considerations further constrain how a tutorial opportunity may appear. If there are only a few actions that a student can take to argue for the value of an ad within the simulation, the system can focus on those particular actions, if it needs to recognize the student's argument.

The expanded RCD is a more detailed description of a tutorial opportunity, but it does not yet take into account knowledge of the recognition process. The next stage of rule generation is the creation of the rule specification. Here the system develops the outline of a recognition procedure for the opportunity, using knowledge of how rules are applied in GuSS and how they interact. The output is a graph in which nodes correspond to rules and connections correspond to dependencies between rules.

The recognition procedure is finally realized through the process of rule creation that produces a set of rules from the rule specification created in the previous step. At this point, the system must use knowledge of how the MOP vocabulary used in SPIEL can be translated into rule clauses expressed in GuSS's vocabulary of internal simulation states. The result is a rule set that implements the recognition procedure.

Of course, SPIEL is recognizing more than one storytelling opportunity at a time. After rule sets have been generated for many RCDs, they can be optimized and combined into a single large recognition procedure in which overlap between rule sets is eliminated. This is the process of rule optimization.

9.2. Expanding the recognition condition description

The recognition conditions in the RCD are extremely general. It does not matter if the student is acting in a real-life role-playing game, the GuSS simulation, or some other environment where Yellow Pages ads are sold; an RCD would still describe an opportunity for storytelling. In this sense, the RCD is an environment-independent representation of a situation that constitutes a tutorial opportunity. It describes conditions that may be realized in different ways in different learning environments. I repeat here the RCD for the "Wife watching TV" story, told using the **Warn about assumptions** strategy:

WHEN the student is in the pre-call phase, and
  the client has a business partner that is also a spouse,
  LOOK FOR the student to exclude the spouse from the discussion, or
    the student to fail to gather information from the spouse.

THEN TELL "Wife watching TV"
AS a "Warn about assumptions" story, other-focused.

The expanded RCD or eRCD is a less abstract version of this description, closer to the kinds of conditions that may arise in the GuSS simulation. The RCD is processed using *elaboration rules* that indicate how abstractions present in the RCD may appear in GuSS. These are similar to manifestation rules used by the storytelling strategies, except that they are specific to GuSS's learning environment.

For example, because YELLO is organized around sales calls, the concept of the student failing to gather information from a spouse in a pre-call context can be recognized by a sequence of two actions. First, there must be a conversation (the primary means of information gathering), and then that conversation must end with the student having failed to gather information. The following elaboration rule encodes this knowledge:

TO RECOGNIZE the student ignoring an opportunity to perform a conversational action with a person.
LOOK FOR the student having a conversation with that person
in which that action could be performed, then
that conversation ending,
without the student having performed the action.

What the rule does is convert a single MOP in the RCD into a recognition sequence. Where the RCD contained "ignore information-gathering," the eRCD contains two events, the beginning and end of a conversation, with the restriction that no information is gathered during it. Other types of elaboration rules may introduce disjunctions or restrictions, or eliminate conditions that can be assumed to hold true. The expanded RCD for "Wife watching TV" is as follows:

WHEN there is a business owner, and
the spouse of the business owner is also a business partner, and
the student is in the pre-call phase,
LOOK FOR
either the student to engage the owner in conversation, then
make a statement excluding the spouse from the conversation, or
the student to engage the spouse in conversation, then
end the conversation,
without having gathered information from the spouse,
THEN TELL "Wife watching TV"
AS a "Warn about assumptions" story, other focused.

9.3.1. Feature detection

Each part of the eRCD is something that can be directly recognized in the simulation, with the exception of one clause, the requirement that the student be in the pre-call phase. How is the system to know where the student is in the sales process? SPIEL could elaborate all of necessary steps to recognize this fact, but almost every storytelling opportunity turns out to require the recognition of the student's progress through the sales call. This feature turned out to be so common that it became useful to create a special feature detector (Owens 1991) for it. SPIEL has a small set of rules that track the student's position in the sales process, maintaining a record that reflects the student's current progress. Any recognition procedure can refer to this value rather than having to gather its own evidence. Therefore, the rule set for "Wife watching TV" does not need to worry about all the possible ways that the student might end up in the "pre-call" part of the sale. It need only check that the "pre-call" recognition rule has matched.

In this tracking mechanism, SPIEL has what could be considered a dynamically-updated student model. It tracks a certain aspect of the student's state using a model of the task and

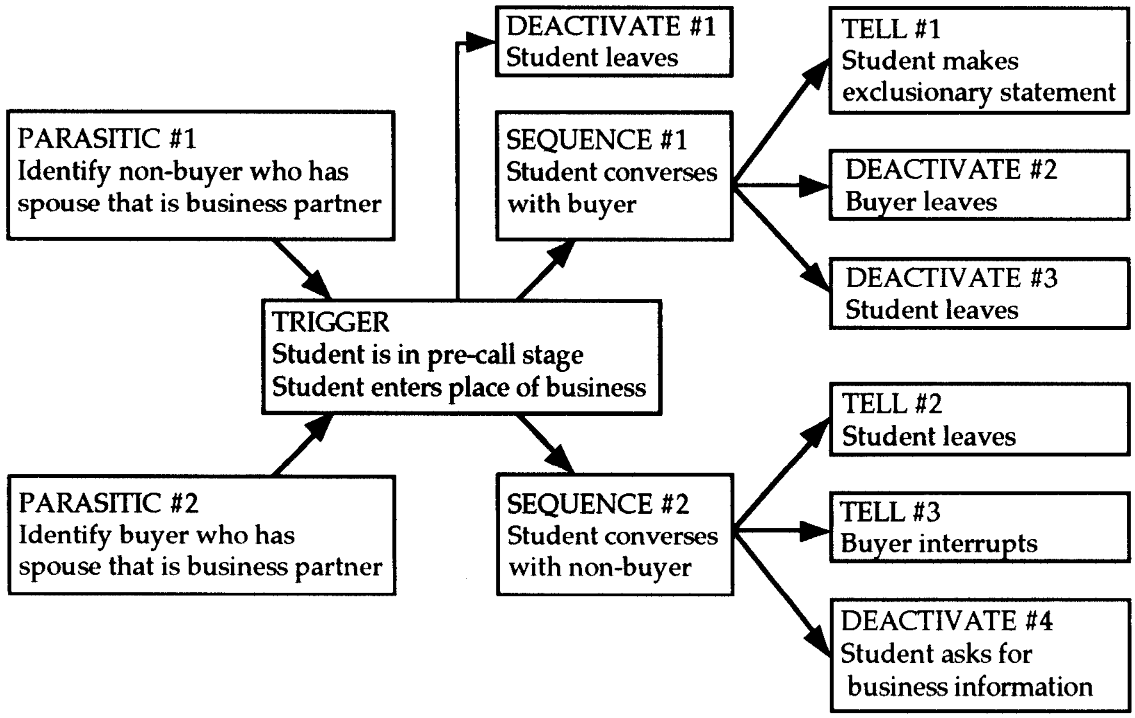

Figure 9.2. Rule specification for "Wife watching TV" told using **Warn about assumptions** strategy

maintains a description of that state for the purposes of retrieval. Although I have argued elsewhere in this dissertation that a complete dynamically-updated student model is not feasible or appropriate as the basis for storytelling intervention, this is a case where it is possible and useful to do a small amount of modeling. It is possible to model the stages of the sales call, because the model is a simple temporal succession, and because there is a simple set of rules for determining when the student is in each state. It is useful to do so because this feature turns out to be needed to recognize nearly every storytelling opportunity.

SPIEL has five sales call tracking rules that detect the student's progress through the selling task. The student always begins the scenario in the **initial** phase. The **pre-call** stage is recognized when customer contact begins. Pre-call ends when the student starts to **prepare** an ad campaign. At the point where the student contacts the customer to sell the prepared ad or ads, preparation ends and **presentation** begins. The system recognizes the **close** as the point in the presentation after the recommended ad program has been presented. These rules are part of SPIEL's rule set and continually update its representation of the student's position in the sales call process.

9.3. Designing the recognition procedure

The eventual output of the storage phase is a set of rules. With the expanded RCD, the system has a more concrete description of what it is looking for, but it does not yet have a recognition procedure. As a first step towards generating this procedure, SPIEL creates what is essentially a flowchart for recognition of the opportunity, showing what must be recognized and when. This flowchart, or *rule specification*, is a directed acyclic graph in which each node stands for a rule.

Figure 9.2 shows the rule specification for the "Wife watching TV" story. It is, for the most part, a simple transformation of the eRCD. Ignoring for a moment the "deactivate" rules, the specification has as one node for each condition in the eRCD (lines in the English description),

Figure 9.3. A redundant rule specification and a simplified version.

with the temporal and logical dependencies in the eRCD captured in the between-node connections.

The deactivating conditions, which are not directly mapped from the eRCD, are present so that SPIEL can deactivate the recognition sequence if the context needed for recognition stops being true or if events in the simulation render the retrieval conditions unachievable. These deactivation rules are necessary if the rule mechanism is to prevent spurious remindings, for example, telling the "Wife watching TV" story if the student has succeeded in finding out business information from Mrs. Swain.

In some cases, the deactivation conditions are explicit in the eRCD. In the example above, the restriction "without having gathered business information..." can be easily turned into deactivating rule #4 that looks for the student to succeed in gathering information and halt recognition if that happens.

In other cases, the deactivating conditions will have to be inferred from other aspects of the eRCD. Suppose the system is looking for the student to make a remark to Mr. Swain regarding Mrs. Swain (telling rule #1), but the student never does. The system recognizes when the conversation ends that it should stop looking for such a statement. This is the role of deactivating conditions #2 and #3. If SPIEL did not have these rules, the telling rule would continue to try to find such a statement even when the appropriate context for storytelling has passed by.

After deactivating conditions have been added, SPIEL makes a correction to eliminate redundancies between its branches: nodes where the same thing is recognized twice. This is more than an efficiency concern: it is a necessity. GuSS's rule mechanism fires only one rule at a time. If there are two rules under consideration at one time, both of which have the same left-hand side, only one will be permitted to fire. Since the situation in the simulation constantly changes, the more redundant rules there are, the more likely it is that the system will fail to recognize a storytelling opportunity because the relevant rule did not get a chance to function. Such duplicate rules can be eliminated by revising the design of the recognition procedure before rule creation occurs. For example, suppose an expanded RCD looks like this:

WHEN A
LOOK FOR
B followed by C followed by D, or
B followed by C followed by E, or
F followed by B followed by G.

The rule specification graph would look like the left half of figure 9.3. Using a process similar to rule optimization, it is easy to determine that the tests for B and C in the first two sequences are redundant and can be merged. (The test for B in the third sequence cannot be shared since no other test for B has the precondition of F.) The right half of the figure shows the result of redundancy elimination.

A second kind of redundancy can occur between parts of a recognition sequence. Although the RCD is always expressed in terms of sequences of conditions, some pieces of a sequence do not require strict ordering. Suppose a rule specification contains a sequence of two conditions: (1) the student asking about the meaning of a term, followed by (2) the client having the expectation that the student would already know the meaning of the term. Because the client's expectation will be present at the same time as the student's action, it is possible to test for both of these conditions in a single rule. SPIEL uses *persistence rules* to check for these cases. Here is an example:

IF A and B are adjacent rules, and
    A is a student action, and
    B is a character's mental disposition, and
    A has no other successor but B,
THEN the merged rule A^B can replace the sequence A followed by B.

9.4. Rules for the GuSS simulation mechanism

GuSS models the simulated agents in its simulated world (including SPIEL and the other tutorial modules) using production systems similar to those found in expert systems (Brownston et al. 1985). At each point in the simulation, an agent looks for the production, or *decision rule*, that best matches its current *mental state* (what it knows, believes, expects, etc.) and fires it. For example, Lucy has a decision rule that causes her to respond to compliments about her house by saying: "Thank you, we like it." This rule fires when she observes the student's comment, "What a wonderful view of the lake you have." Other rules cause changes to her mental state: she stops expecting Mike to arrive when she greets him, for example.[1]

9.4.1. Agents as planners

Each agent in GuSS is an independent planner, taking input from the simulation and responding in a manner designed to achieve its goals. The internal states of such a planner therefore represent intentions, such as goals and plans, and beliefs that support them. In the GuSS simulation, an agent's mental state is represented as a list of assertions of what the agent knows, feels, or intends. The list is ordered, reflecting a simple notion of priority: those at the top of the list being more likely to influence processing. New states that arise are added to the top. Each agent's initial mental state is determined by the designer of the scenario when the initial conditions are established.

The production system-based agents interact with each other and the user in the GuSS simulation engine, a simple event-processing loop. Agents (and other objects in the simulation) respond to events, things that happen in the simulated world, as they are pulled from the *event queue*, a queue of the events that are scheduled to occur in the simulation. Agents may respond by generating new events that are in turn placed back on the queue. For example, an agent who processes the event of hearing a phone ring may make a decision to answer it, generating a new event of picking up the phone. The agent on the other end of the line will notice this event and respond appropriately. When the student performs actions using the action constructor, each action becomes an event that is placed on the event queue.

9.4.2. GuSS's decision rules

Like production rules, GuSS's decision rules are condition/action pairs. However, the condition part of each rule is divided into three parts: retrieval conditions, filtering conditions, and discrimination conditions. Retrieval conditions are used to access each agent's table of decision rules. An agent makes a decision by first using its mental state to key into the table. It retrieves a set of candidate rules, each of which has retrieval conditions compatible with the

---

[1]See (Blevis et al. in preparation) for a detailed account of the design and implementation of the GuSS simulation. The simulation is also described in (Kass et al. 1992; Kass et al. to appear).

Rule #1:
If

(retrieval conditions)	I observe someone enter my office, and
(filtering conditions)	I am by myself, and
(discrimination conditions)	I'm expecting that person to come at a certain time, and it's the right time, then
(actions)	I will say "Hello," stop expecting the person to arrive, and start believing that the person has arrived on time.

Rule #2:
If

(retrieval conditions)	I observe someone enter my office, and
(filtering conditions)	I am by myself, and
(discrimination conditions)	I'm engaged in an activity, and I expect that people will not interrupt me, then
(actions)	I will say "What?," become somewhat angry, and start to believe that the person is rude.

Rule #3:
If

(retrieval conditions)	I observe person A enter my office, and
(filtering conditions)	I am conversing with person B, and
(discrimination conditions)	I'm expecting person A to come at a certain time, and it's the right time, then
(actions)	I will say "Hello," stop expecting the person to arrive, and start believing that the person has arrived on time, and come to have the goal of ensuring that person A is acquainted with person B.

Figure 9.4. Three decision rules with the same retrieval conditions.

agent's current mental state. Because an agent's entire mental state can be quite large, on the order of several hundred SOMs, GuSS only permits a subset of the mental state to be used for retrieval conditions: only goals, plan-steps, observation, and members of a special class of belief called awares.

For example, suppose Ed Swain observes someone enter his office. He may have several rules whose retrieval conditions are compatible with this situation. The rules in figure 9.4 would all be retrieved as possible candidates.

The filtering conditions are employed to filter the candidate decision rules chosen by the retrieval conditions. There are "only when" conditions that must match for the rule to be executed, and there are "not when" conditions that, if matched, cause the rule to be rejected. If Ed is meeting with someone, he will reject rules #1 and #2 and use #3.

The discrimination conditions are used to rank candidates if there are several that pass the other tests. They are matched against the agent's state of mind using a match algorithm that returns a numerical score. The highest-scoring decision rule is executed. If Ed Swain were alone and expecting a visitor, Rule□#1 would match better than #2 and would be executed.

There are a few major differences between GuSS's agents and common production system models, such as OPS5 (Brownston et al. 1985). Most important, GuSS uses retrieval conditions to perform retrieval prior to the matching step. The matching step therefore occurs only on a small set of rules, the ones that are retrieved because they are likely to be relevant to the agent's current situation. This means that working memory and the set of productions can be larger without incurring much additional pattern-matching. Giving Ed Swain new topics to discuss will not slow down his existing responses greatly, because the rules that implement his conversation about one topic will not be retrieved when the conversation is about another topic. Another advantage to the two-level matching scheme is that generic rules can easily be overridden by agent-specific ones and still be available as fallback options.

Another important aspect of GuSS's decision rule mechanism is its conflict resolution. Discrimination among decision rules occurs on the basis of the qualitative measure returned by the pattern matcher. GuSS uses a hierarchy of concepts when performing its matching, giving greater weight to matches that are specific. A rule containing an abstraction will match a mental state that contains a more specific concept, but not as strongly as a rule that matches exactly. The matcher also prefers matches that do not involve unbound variables. Using the example rules given above, if Ed is both expecting an appointment and engaged in an activity, Rule □#1 would match better, because the expectation of the appointment, containing a reference to a particular person, is more specific than the discrimination conditions in Rule □#2. This notion of specificity is different from the measure of rule specificity in OPS5, which counts the number of conditions in a rule's left-hand side as the measure of how specific it is.

Since SPIEL's retrieval-time component is implemented as an agent in the GuSS simulation, it participates in the central cycle of observation and action, just like the characters with whom the student is interacting. However, the SPIEL agent has special properties. As indicated in the chapters on storytelling strategies, SPIEL must occasionally inspect the mental state of other agents, something that no character is permitted to do. Also, as described later in this chapter, SPIEL attaches decision rules to other agents in the simulation, so-called *parasitic rules*, to perform certain computations. The SPIEL agent also requires additions to the GuSS vocabulary to create the unique linking states that tie a rule set together.

9.5. Rule creation

In the rule specification, each node represents some state or conjunction of states that must be recognized at a particular point in the recognition sequence. The task of the rule creation step is to create production rules that will recognize these states within the GuSS simulation.

There are two primary problems associated with this task. The most obvious problem is that of vocabulary. The states expressed in the rule specification are essentially in the same vocabulary as the index, that is, MOP structures. The states against which rules must match in the GuSS simulation are in a different vocabulary. SPIEL must convert its recognition procedures into GuSS's language. The second problem is communication between rules, to preserve the structure of the recognition procedure and maintain consistency. (This issue was touched on briefly in the discussion of linking states in Chapter 4.)

9.5.1. Translation of vocabulary

GuSS's representation of states and actions is quite different from that used in SPIEL. MOPs are essentially frame-based representations in which packaging relationships are very significant, while in GuSS, each agent's working memory is represented by a list of assertions, states of mind, or SOMs. An SOM cannot package or point to others. They are always of the following form:

(predicate (agent act aspect object)).

For example, an observation of the student asking about plans for the expansion of Swain Roofing would be as follows:

(observe (student ask-about (plans-for expansion-of) swain-roofing)).

The predicate part indicates what kind of mental state the SOM represents. There are eight types: observations, goals, plan-steps, expectations, beliefs, awares, attitudes and emotions. The agent is the person that the mental state is about. The act is the action that the SOM

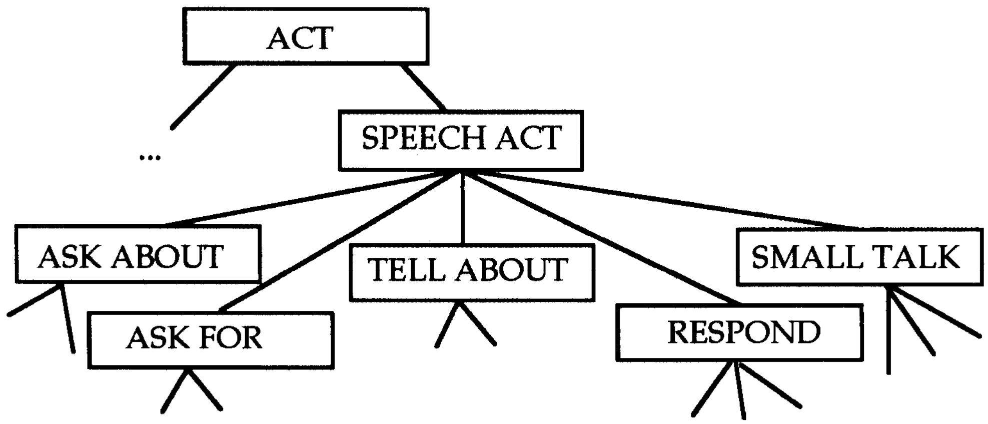

Figure 9.5. A fragment of GuSS term hierarchy.

asserts. Aspects are modifiers that operate on the object of the SOM. Acts are the verbs of the SOM language, aspects are its adjectives, and objects and agents its nouns. In the example, the object is "Swain Roofing," the business owned by Mr. Swain. Multiple aspects in an SOM are composed to yield its meaning, as in the example.

Acts, as well as other terms in GuSS's vocabulary, are drawn from a large representational hierarchy. They are related to each other using isa relationships. "Ask about," for example, is a type of "speech act" which is a type of "act." Figure 9.5 shows a fragment of the term hierarchy for acts in GuSS.

A single SOM is obviously limited in its ability to express structured concepts since acts and objects are simple terms and the only type of embedding permitted in aspects is simple conjunction. For example, it is not possible to represent an assertion that contains disjunction such as "John believes that either Sue or Mary wrote the letter." Complex mental states must be broken into smaller assertions to be represented.

Consider the recognition that the student has engaged the owner's spouse in conversation. The rule has to recognize the student speaking, the spouse being present, and the owner being absent. (In GuSS's world, everyone who is at a given location is automatically part of the conversation, so the only way the student could talk just to the spouse would be if the owner were absent.) SPIEL has a table of translation rules that enables it to convert MOPs into SOM equivalents in decision rules. Here is the translation rule for a conversation involving someone who is not the buyer:

TO RECOGNIZE a conversation between the student and someone other than the buyer,
A RULE NEEDS
RETRIEVAL CONDITIONS: the student says something;
ONLY WHEN: the person who is not the buyer is present;
NOT WHEN: the buyer is present.

The retrieval conditions look for the student to make some speech act. The "only when" verifies that the non-buyer person is present in the room, and the "not when" condition prevents the rule from firing if the buyer is present.

9.5.2. Parasitic decision rules

One unusual feature of GuSS's rule matcher is that it lacks the capacity to backtrack. This creates problems for SPIEL in searching for characters with certain characteristics. In evaluating the retrieval, filtering and discrimination conditions of a rule, GuSS's matcher operates on one condition, one SOM, at a time. It searches the agent's mental state for any matching SOMs. If the rule contains a variable, that variable will be bound for the rest of the evaluation of that rule to the corresponding value in the first matching SOM in the agent's mental state.[2] If the matcher fails when attempting other matches using that variable binding, it does not go back and try other possible bindings.

An example will make this feature more clear. To tell the "Wife watching TV" story, SPIEL must recognize a married buyer, which is a conjunction of SOMs: "AGENT1 is married to AGENT2" and "AGENT1 owns COMPANY1." For each SOM in the rule, the rule matching algorithm scans the agent's mental state looking for a match. Suppose that SPIEL's mental state contains the following three SOMs in order: "Lucy Swain is married to Ed Swain," "Ed Swain is married to Lucy Swain," and "Ed Swain owns Swain Roofing."

The matcher starts with "AGENT1 is married to AGENT2," and finds "Lucy Swain is married to Ed Swain." This matches with AGENT1 = "Lucy Swain" and AGENT2 = "Ed Swain." Then, it will try to match "AGENT1 (Lucy Swain) owns COMPANY1," but fail to find a match since Mrs. Swain does not own a business. Rule application will fail, even though there is a variable assignment, AGENT1 = "Ed Swain," AGENT2 = "Lucy Swain," and COMPANY1 = "Swain Roofing" that does satisfy the conjunction.[3]

This limitation of the pattern matcher means that a rule containing a variable cannot be guaranteed to operate correctly unless (1) there is only a single possible match in the agent's mental state or (2) the rule does not involve conjunctions of conditions using the same variable. GuSS's simulation rules are usually written this way. SPIEL, on the other hand, must frequently select a character with a conjunction of characteristics out of several that may be present in a scenario. SPIEL needs to be sure that if there is a character with the desired characteristics it can latch onto it.

To see how SPIEL gets the desired behavior out of GuSS's rule application mechanism, it is useful to recognize that backtracking is merely the most common implementation of non-deterministic choice. If a matching algorithm needs to make a choice between possible variable bindings, it can be guaranteed to find the right one if it makes a choice while remembering the bindings it did not choose, so they can be tried if the selected ones fail.

Another implementation of non-deterministic choice is to pursue all choices in parallel: if there is a correct binding, then one of the parallel matches will succeed. Since agents are the objects between which SPIEL must choose when making bindings, and since agents are allocated their own processing time (essentially in parallel) to process events, SPIEL can piggyback on each agent, getting them to evaluate their own characteristics.

Whenever SPIEL needs to identify a character with certain properties, such as a business owner, it creates a *parasitic decision rule*.[4] These rules, while created by SPIEL and serving to help the story retrieval process, actually exist in the decision rule tables of the characters in the simulation. As their name suggests, parasitic rules steal a small amount of processing from each agent, usually only once at the beginning of the scenario. The rules inspect the agent to

---

[2]In GuSS, variables will only appear in rules. They are not permitted in an agent's mental state.

[3]I anticipate that future versions of GuSS will incorporate a backtracking matcher, rendering this point moot. Backtracking will probably become necessary when the simulation is called upon to handle scenarios of increasing complexity, where a multiplicity of possible variable bindings becomes the rule rather than the exception.

[4]I am indebted to Tom Murray for the term "parasitic" and the realization that parasitic decision rules could be viewed as a solution rather than a problem.

```
;; PARASITIC D-Rule <SPIEL-PARA-DRULE-1952> for story: "Wife watching TV"
;; For AGENT
;; Vars: ((*SELF #<agent>)
;;        (*AGENT1935 #<agent>)
;;        (*AGENT1936 #<agent>))
;; Indices:     ((AWARE (*SELF #<know about> #<anything> #<time>)))
;; Only-when: ((BELIEVE (*SELF #<is spouse> #<anything> *AGENT1936))
;;              (BELIEVE (*SELF #<is business partner> #<anything> *AGENT1936))
;;              (EXPECT (*SELF #<possess> #<control of> #<business>)))
;; Not-when:  <none>
;; Peek:        <none>
;; Not-peek:  ((AWARE (SPIEL-AGENT #<linking som 1939> #<anything> *AGENT1936))
;;              (AWARE (SPIEL-AGENT #<linking som 1938> #<anything> *SELF)))
;; Agent peeked: SPIEL-AGENT
;; Poke:        ((AWARE (SPIEL-AGENT #<linking som 1939> #<anything> *AGENT1936))
;;              (AWARE (SPIEL-AGENT #<linking som 1938> #<anything> *SELF)))
;; Agent poked: SPIEL-AGENT
;; Action:    <none>
;; Forget:    <none>
;; Learn:     <none>
```

Figure 9.6. Parasitic decision rule for "Wife watching TV" story

which they are attached, determining if the agent meets the necessary criteria. They signal the teaching module if their character meets the desired conditions. In this way, SPIEL can determine which characters have the desired characteristics.

Figure 9.6 shows an example of a parasitic decision rule used for the "Wife watching TV" story. To explain this rule, I will have to discuss a few more details of GuSS's rule application mechanism. The retrieval conditions are called (somewhat confusingly) *indices*. I will continue to use the term retrieval conditions to avoid confusion with SPIEL's indices for its stories. In this rule, the retrieval conditions correspond to the agent's on-going awareness of time, something which is always present from the beginning of the simulation. This rule will therefore be retrieved as soon as the simulation starts up.

This rule, like all parasitic rules, belongs to all agents. All characters in the simulation decide whether or not they meet the criteria of the rule and pass that information to SPIEL. Because its retrieval conditions are so general, this rule will always be retrieved. The rest of the conditions are filtering conditions that determine whether, for a given agent, this rule will be fired. First are the "only when" conditions, all of which must be true in order for the rule to fire. For this rule, the agent must believe that it has a spouse and that the same person is also a business partner. It also must expect to be in control of a business, a state that marks a character as a business owner in YELLO. These are conditions that would hold true of Mr. Swain. The rule will also be retrieved for Mrs. Swain and Dave Swain, but Mrs. Swain fails the "ownership" test and Dave fails the "spouse" test.

For Mr. Swain, then, this rule will be retrieved and the "only when" conditions will be met. Successful match means that the action part of the rule is performed, in this case the "poke" action. It places two linking states in the SPIEL agent's state of mind as a record that this match has been successfully performed.

Since he will retain his ownership of Swain Roofing and his marriage throughout the scenario, there is nothing in the conditions seen so far to prevent Ed from firing this rule at every time step, filling the SPIEL agent's mental state with new copies of the same information. To prevent such an inefficiency, the rule must actively prevent itself from firing again after it has fired once. This is the reason for the "not peek" conditions in the rule, which are analogous to the "not when" conditions, except that they reference the mental state of another agent, in this case the SPIEL agent. The rule tests to see that the linking states that it gives the SPIEL agent are not already present. If they are, the rule has already fired and it can be ignored.

9.5.3. Linking states and variable binding

As described in Chapter 4, SPIEL's rules communicate with each other through the use of linking states. Rules need to coordinate when they are connected in a temporal relation to each other. The predecessor rule needs to tell its successors, "I've fired," so that the successors can begin trying to recognize their piece of the recognition sequence. The "I've fired" message comes in the form of a unique SOM that the predecessor rule adds to the SPIEL agent's mental state. Rules that follow use the presence of this state as a filter. They are inhibited from firing until the predecessor fires and inserts the linking state. A successor rule will remove the linking state upon firing so that only one recognition path is followed.

Linking states are the synchronization mechanism for predecessor and successor rules, but they also serve two other purposes that are apparent in the parasitic rule example: (1) they block the repeated firing of the rule that created them; and, (2) they record important variable bindings. For parasitic rules in particular, this last function is important. The only reason for having these rules is to obtain correct variable bindings, since the rule matcher cannot be guaranteed to do so. These bindings are communicated to the rest of the recognition procedure using the linking states. In fact, rules of all types frequently must communicate values to each other in this way. If a context depends on Ed Swain not being in a certain location, that location must be bound and passed along so that future rules will be able to detect if Ed has entered that location.

The communication of variable bindings can be seen in the two linking states created by the parasitic rule shown above, one containing the person whom the rule matches and one containing the spouse of that person. If a later rule needs to know who the owner's spouse is, it needs to have a condition of the following form:

```
(AWARE (SPIEL-AGENT {UNIQUE LINKING ACT} #<anything> *VARIABLE))
```

The "act" part of the SOM is unique to the particular rule that created the linking state. This condition will only match the SOM created by that rule in which the correct binding can be found. Therefore, VARIABLE will be bound to the owner's spouse.

The parasitic rules recognize characteristics of people that do not change over the course of the scenario. Ed and Lucy will not divorce while the sales call is happening. The rules need only detect their properties once. These linking states are therefore permanent, not removed by other rules. Any rule with a question about an agent's characteristics can consult the linking states created by the parasitic rules. On the other hand, most linking states communicate information of transitory value, such as "the student has just walked into the Swain's kitchen." Each of these linking states is a trigger for some following rule, which removes the state from SPIEL's mental state upon firing. The binding in such states cannot be universally relied on.

Therefore, it may happen that a rule binds a variable and passes it to the next rule, a rule that binds a different value and communicates it. A later part of the recognition procedure may need to refer back to the earlier value, but it is now forgotten. For example, suppose we have the following simple rule set.

Rule A looks for the student to create an ad with certain properties;
Rule B looks for the student to go to the client's place of business;
Rule C matches against the student's action of presenting the ad to the client; and
Rule D deactivates the recognition if the student leaves without presenting the ad.

Suppose the first two rules match successfully, as shown in figure 9.7. What value should Rule B pass along in its linking state? Rule C wants to know the identity of the ad to look for

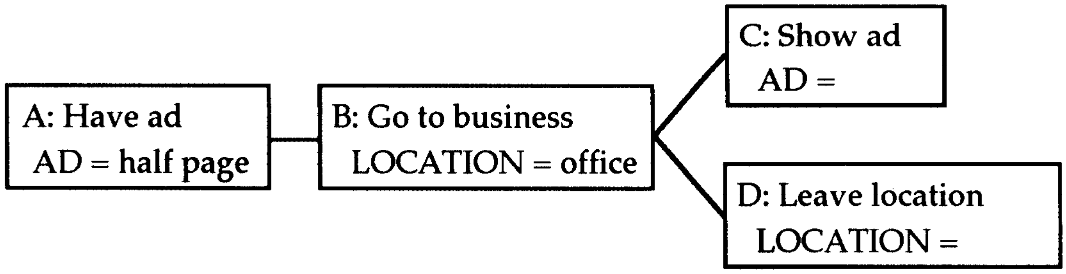

Figure 9.7. Communication of values within a rule set.

the presentation of it. It cannot refer back to the linking state produced by Rule A, since that state was removed by Rule B. Rule D needs to have the value of the location so it can look for the student leaving that location.

To satisfy the needs of both of its successor rules, Rule B must produce two linking states to pass both values along. The rule creation algorithm takes note of the values that each rule needs to test its conditions and checks to see whether these values have been previously bound in the recognition procedure. If they have, the rule generator adds linking states to all of the rules between the original binding and the place where the value is needed to create a bucket-brigade that passes the value along.[5]

9.6. Rule optimization

Rule optimization was outlined in Chapter 4. With a greater understanding of the rule generation process, it will now be easier to give a detailed account of optimization.

The rule generation process creates, for every recognition condition description, a rule set that implements a recognition procedure for the RCD's tutorial opportunity. The rules, as I have shown, are tightly coupled with each other through linking states and variable bindings. However, there are many redundancies between rule sets. This occurs most frequently in the case of triggering conditions. Many stories require that the system identify who "the buyer" is and watch how the student interacts with that person. Many stories can only be told in a particular phase of the sales call, such as the sales presentation.

As was the case for rules within a recognition procedure, it is desirable wherever possible to merge redundant rules, ones that have the same predecessors and the same retrieval, filtering and discrimination conditions. Optimization proceeds by moving forward through all of the rule sets simultaneously. It looks first at rules that do not depend on others, ones that initiate recognition sequences. This will usually be the parasitic rules. It partitions the group of rules based on redundancies. All rules in the same partition are replaced with a single rule that performs the same test. The process then moves along to the rules that follow those just optimized, until there are no more rules left.

SPIEL optimizes by walking through the recognition procedures from left to right, preserving their linear, sequential structure, but consolidating across procedures where possible. Its criteria for redundancy are fairly strict: the left-hand sides must be identical (except for linking state information) and the predecessors must be the same. A rule that is three steps along in the recognition chain can only be merged with another one that is three steps into the same chain. For the optimizer to realize that the rules are in fact steps along the same chain, it must have already optimized their predecessors.

---

[5]In the current implementation of rule generation, linking states have only been called upon to pass the values of objects between rules, never aspects or acts. Other types of values could easily be communicated in a similar way.

So far throughout the rule generation process, each tutorial opportunity has kept its distinct identity: each RCD has an expanded RCD, a rule specification, and a rule set. However, once the final stage of optimization is complete, the rule sets are no longer distinguishable. The whole collection of rules has been merged to create a opportunity recognition machine for all of the storytelling opportunities.

Optimization operates on the whole collection of rule sets, which may contain as many as several thousand rules. For illustration, I will show a miniature example using two rule sets. Consider the two rule specification graphs shown in figures 9.8 and 9.9. They are for the rule sets that recognize opportunities to tell the "Taxidermist" story using the **Warn about pessimism** strategy, other focused, and "Obnoxious attorney" using the **Explain other's plan** strategy. Although optimization occurs on the rules themselves, it is easier to see the optimization process with the graphical representation, the rule specification.

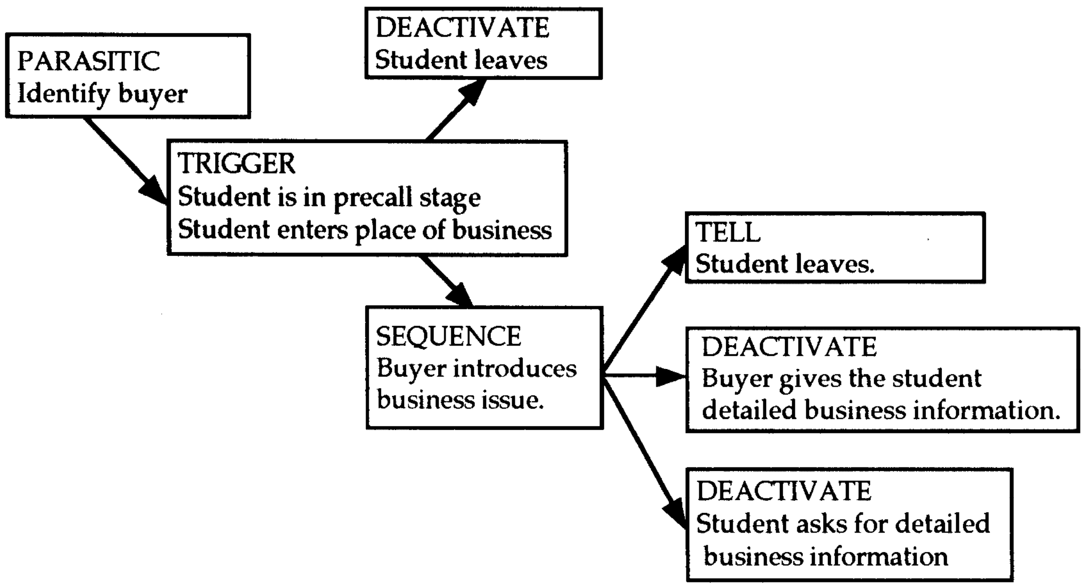

Figure 9.8. Rule specification for telling "Taxidermist"
using the **Warn about pessimism** strategy, other focused.

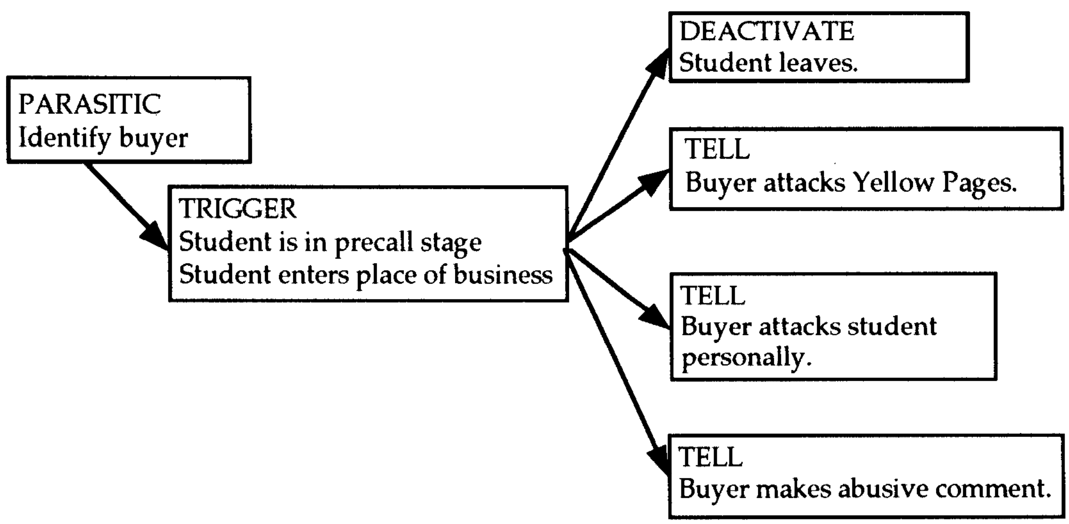

Figure 9.9. Rule specification for "Obnoxious attorney"
using the **Explain other's plan** strategy.

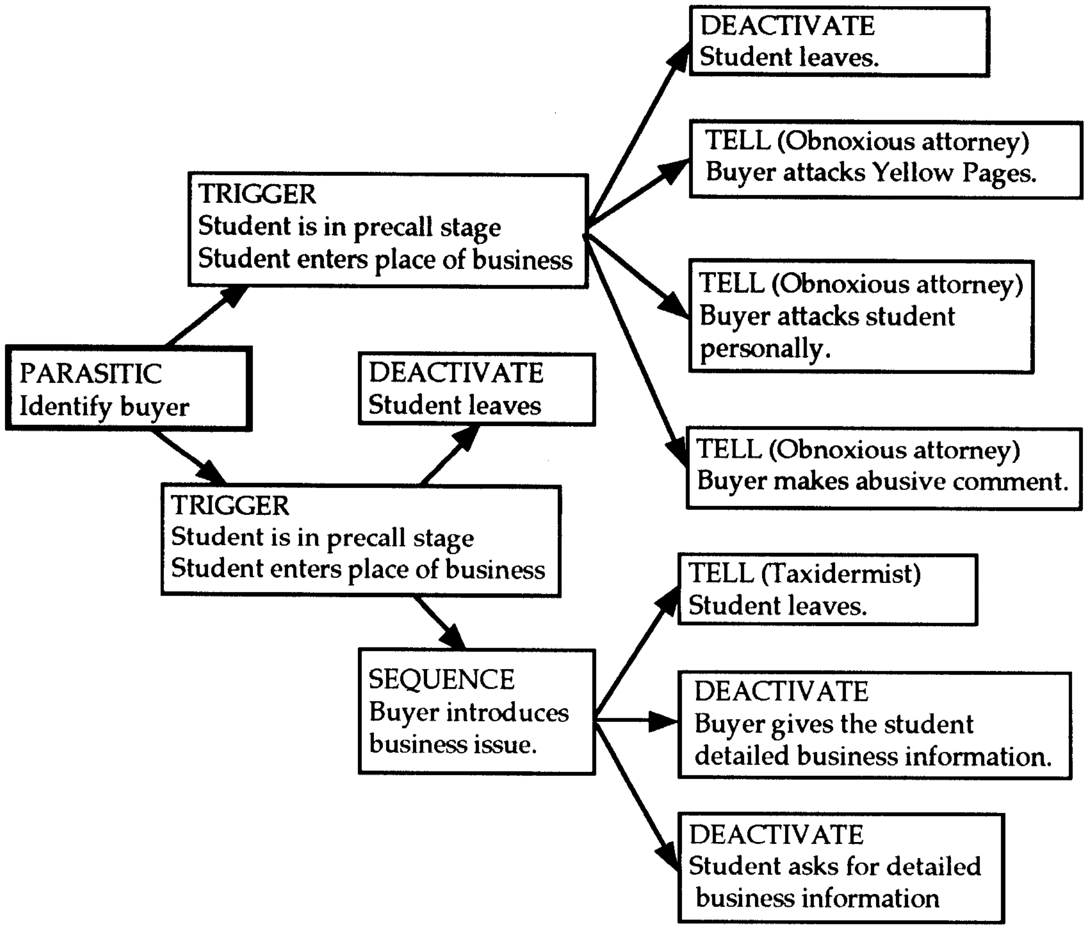

Figure 9.10. Two rule sets after one stage of optimization.

Because both parasitic rules recognize the same thing, the presence of the buyer, they can be optimized in the first optimization step. A new rule is created with the same retrieval and discrimination conditions, a test for "buyer-hood," and a new action passing along a new linking state. The rules that come after these parasitic rules in their respective rule sets are then adjusted to look for the linking state created by the new rule, effectively drawing the new rule into the recognition sequence.

The rule sets are shown in figure 9.10 after the first optimization step. The darker box surrounding the parasitic rule shows that it has been optimized. Now the two triggering rules are inspected. They perform the same recognition function and have the same predecessor, and are therefore merged in the second step.

Following the triggering rules, each rule set has a deactivating rule that looks for the student to leave. They are consolidated in the third step of the optimization. At this point, the recognition procedures diverge and no more optimization is possible. The final rule set is shown in figure 9.11. Three rules have been eliminated, but the recognition sequences for both tutorial opportunities have been preserved.

9.7. Rule application

The storage time processes in SPIEL build rule-based recognition procedures. At retrieval time, a student interacts with a GuSS scenario and the SPIEL agent attempts to use its recognition machinery to retrieve stories. This section shows a trace of how the "Wife

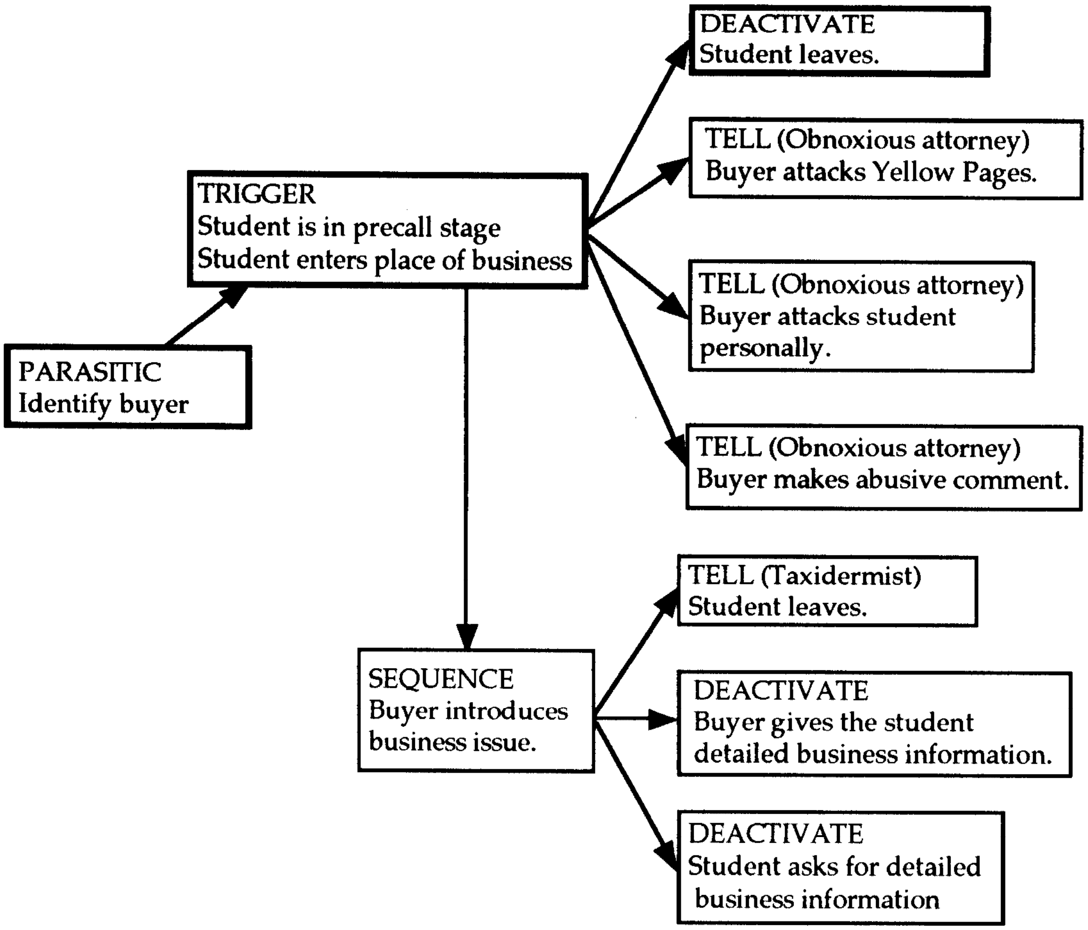

Figure 9.11. Two rule sets after completion of optimization.

watching TV" story is retrieved. Again, simplification has been necessary. Recognition actually takes place using the optimized rule collection in which no single storytelling opportunity has a distinct recognition procedure. Rules do not "belong" to a particular story at this point. However, for the purposes of exposition, the "Wife watching TV" recognition procedure is treated as through it were operating on its own. Figure 9.12 again shows the rule specification for this story.

This is a fairly complex rule set. There are two parasitic rules, one that looks for the spouse that is the owner and another that looks for the one that is not. After these characters have been identified and the pre-call stage has begun, there are two possible recognition paths, corresponding to the disjunction in the original RCD. Either the student will engage in conversation with the owner or with the spouse. (If the student does not talk to either and leaves the pre-call phase, then the recognition process halts.) If the student talks with the owner, the story gets told if the student explicitly excludes the other spouse from the conversation.

The second possibility is that the student will talk to the non-owner spouse. Then, the story gets told if the conversation ends with the student having failed to take advantage of the information gathering opportunity. This is what happens in the example in Chapter 2.

The first parasitic rule was examined in detail in the rule generation section. I will describe the function of the rest of the rules here without going into their implementation. The full implementation of this rule set can be found in Appendix A.

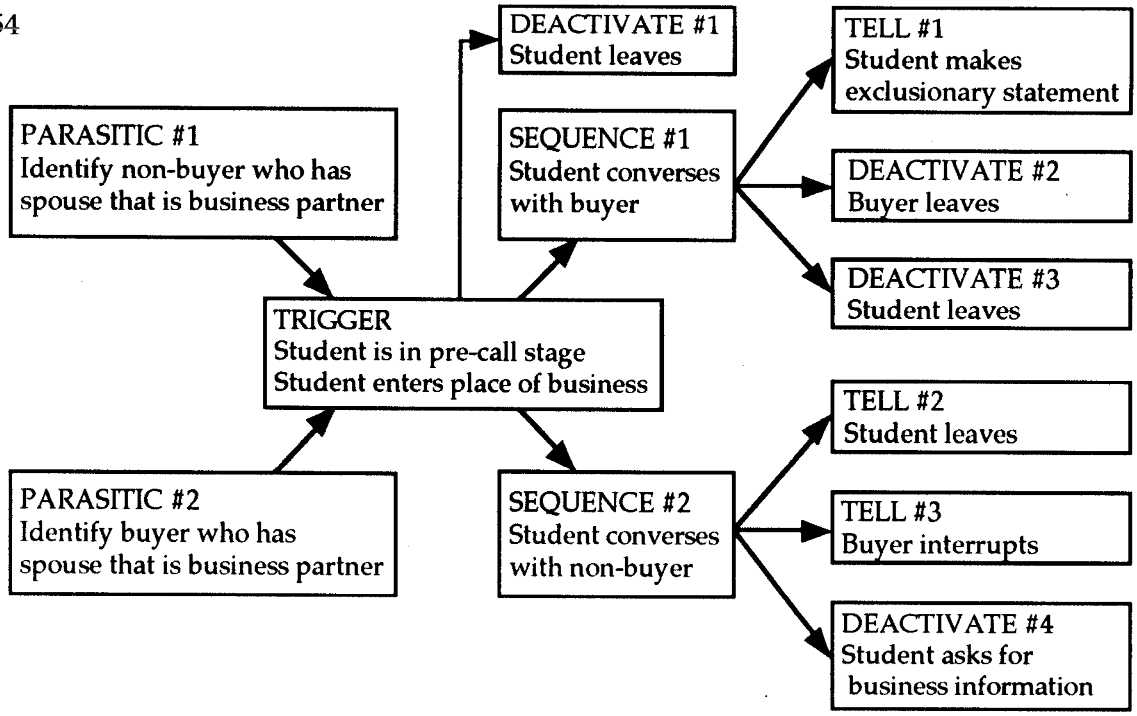

Figure 9.12. Rule specification for "Wife watching TV" told using **Warn about assumptions** strategy

The second parasitic rule matches Mrs. Swain, since she is not a business owner. Figure 9.13 describes the mental state of the SPIEL agent after the simulation has started and the two parasitic rules have had a chance to put linking states into the SPIEL agent's mental state. These four linking states serve to communicate the structure of the relationship between Mr. and Mrs. Swain.

The next rule is a triggering rule. It verifies that Mrs. Swain is both business partner and spouse to the same person that has her as a business partner and spouse: both directions of the relationship are confirmed.

The triggering rule does not remove the linking states that are due to the parasitic rules, since they will remain true throughout the course of the simulation and may be referred to by other rules. It does add its own linking state for its successors to refer to. Because rules in this sequence depend on knowing the location where the pre-call is taking place, this linking state records the location that the student is in. Figure 14 shows the SPIEL agent's state following this rule firing. Note that the rules responsible for maintaining the record of the student's

Parasitic rule #1 matched against "Lucy Swain"
Parasitic rule #1 matched against spouse & bus. partner "Ed Swain"
Parasitic rule #2 matched against "Ed Swain"
Parasitic rule #2 matched against spouse & bus. partner "Lucy Swain"
Student is in the initial stage of the sales call.

Figure 9.13. SPIEL agent's mental state after firing the parasitic rules.

Triggering rule matched with location "Swain's kitchen."
Student is in the pre-call stage of the sales call.
Parasitic rule #1 matched against "Lucy Swain"
Parasitic rule #1 matched against spouse & bus. partner "Ed Swain"
Parasitic rule #2 matched against "Ed Swain"
Parasitic rule #2 matched against spouse & bus. partner "Lucy Swain"

Figure 9.14. SPIEL agent's mental state after firing triggering rule.

progress through the sales call have recorded the fact that the student is now in the pre-call stage. SPIEL is primed to look for the storytelling opportunity.

The next important event in the simulation is that the student begins talking to Mrs. Swain. This signals the information-gathering opportunity that the recognition conditions refer to. This fact is recognized by sequence rule #2. It picks the binding for the two spouses from the linking SOM and checks for the presence of the non-owner spouse and the absence of the owner (if Ed were there it would count as a conversation with him, not with Lucy).

Having recognized the initiation of the conversation, the sequence rule removes the linking state contributed by the triggering rule, and replaces it with one of its own. This will prevent the firing of either of the other two rules that are successors to the triggering rule, deactivation rule #1 or sequence rule #1. See figure 9.15 to see what the SPIEL agent knows at this point. Although the sequence rule does not actually use the location value passed to it, it continues to transmit it to the rules that follow since they look for the student's location.

Now the question is "Will the student find out what Lucy knows?" The story will only be told if the student fails in information gathering. What happens is that Ed arrives and the student is still talking about fishing. The information gathering opportunity has been missed. The telling rule (#3) that notices this is quite simple. It just looks for the owner arriving, then removes the last linking state, and signals the interface to recall and present the story.

The reason is that the telling rule can be so simple is that there is a deactivating rule (#4) that looks for the student gathering business information from Lucy. If that rule had fired, the linking state from the sequence rule would have been removed and the telling rule would never have been retrieved. Therefore, having made it this far with the linking state still present, the system knows that the student did not take advantage of the opportunity, and the "Wife watching TV" story is therefore relevant.

9.8. Retrieval in SPIEL

This chapter has offered a tour through the innermost details of SPIEL's rule-based retrieval

Sequence condition #2 matched, passes location "Swain's kitchen"))
Student is in the pre-call stage of the sales call.
Parasitic rule #1 matched against "Lucy Swain"
Parasitic rule #1 matched against spouse & bus. partner "Ed Swain"
Parasitic rule #2 matched against "Ed Swain"
Parasitic rule #2 matched against spouse & bus. partner "Lucy Swain"

Figure 9.15. SPIEL agent after recognizing the conversation.

mechanism. This discussion completes our examination of SPIEL's story retrieval process. SPIEL starts with a library of stories and computes the set of recognition conditions for opportunities to tell them. It is the job of the retrieval mechanism described here to find these opportunities in the midst of the student's interaction.

SPIEL's rule-based recognition method is an implementation of the general idea of retrieval as opportunity recognition, which is an alternative to the standard cue-based retrieval paradigm. There is nothing in SPIEL's retrieval that corresponds either to the cue formulation step or the memory search step found in cue-based retrieval. SPIEL has more in common with systems for opportunistic planning and acting (such as RUNNER [Hammond 1989]) than with information retrieval systems or most case retrievers. This chapter has addressed some of the important issues found in any opportunistic system. Most important is the issue of matching: how an abstractly-characterized opportunity can be compared against a specific situation to determine if an opportunity exists. In SPIEL, this is accomplished for the most part at storage time by using elaboration knowledge to create concrete descriptions of each opportunity that can be directly matched against the simulation. An opportunistic system must also be capable of looking for many opportunities at once. SPIEL's rule generation and optimization mechanisms create a network of rules in which multiple storytelling opportunities share in the accumulation of evidence.

10. Conclusion

10.1. Case-based teaching

SPIEL adds to the small but growing body of research in case-based teaching systems, which includes DUSTIN (Ohmaye 1992) and Creanimate (Edelson 1993). It is interesting to note that each of these systems uses a different kind of case. DUSTIN shows re-enactments in which a situation is acted out in a realistic context. In Creanimate, clips from nature videos are shown. SPIEL uses videos of expert practitioners telling stories of their own experiences.

These techniques are different means of communicating cases. DUSTIN's re-enactments get students to experience a case as spectators watching someone else perform. In Creanimate, the narration in a video often does not provide anything case-like, focusing instead on describing the features of animals, but the videos of animals in action are something else: they provide vivid and detailed experiences that students will readily store and recall.

SPIEL's first-person stories attempt to give students a second-hand version of actual experiences in the social world. Not only do students hear about events from a participant's perspective, but the personal values and beliefs of experts, and by extension the community of expertise inevitably show up in stories (Orr 1990). Stories that are first-hand reports of actual events communicate experience in a case-based form, but allow it to be filtered and distilled by the expert's perspective. It is not always possible to have cases of this type. There is no community of experts who engage in the fanciful animal-design task that students perform in Creanimate. There might, on the other hand, be some utility in including in DUSTIN stories told by non-native speakers who have managed to learn English and who can describe some of their earlier difficulties.

The goal of every case-based teacher is to capitalize on students' natural case-based learning. This can be done by giving students rich experiences that they will remember, as Creanimate does by using striking depictions of animal behavior, or it can be done by describing, in a rich manner, the experiences of others. It is important to guard against the use of "cases" that are really descriptions of general principles or methods, unless such descriptions can be supported by detailed examples, as GuSS's Analyzer does using re-enactments. Impoverished examples lead to impoverished indexing, and diminished chances for learning.

10.1.1. Evaluating case-based teaching

Because case-based teaching systems attempt to help students acquire the case library they need as the basis for case-based reasoning, such systems should be a good test of the case-based reasoning model. If teaching students cases does help them build up their expertise in a domain, then both the teaching method and the reasoning model can be confirmed. The experience with YELLO has shown that students certainly respond to SPIEL's first-person stories and consider them a valuable part of the interaction.

However, there are pitfalls in trying to assess today's case-based teaching systems, SPIEL included. The most important feature of a case-based teacher is not merely that it presents cases, but that those cases are integrated with a compelling instructional environment and presented at just the right times. It is also important that a tutor's presentation strategies correctly estimate students' interest, so that the cases it presents are perceived as relevant. Finally, the tutor must help the student understand the relevance of retrieved cases through appropriate bridging comments. It is difficult to evaluate the value of a case-based teaching system without a greater understanding of the issues of learning environments, relevance assessment, and case explanation. These are the major topics along which research in case-based teaching must continue.

10.2. The indexing problem

This dissertation follows a long line of research devoted to the indexing problem (Schank 1982; Kolodner 1984; Hammond 1986; Owens 1990; Domeshek 1992). Perhaps Domeshek posed the question best: "If systems are intelligent primarily through access to relevant past experience, how is such access to be assured?" (Domeshek 1992, 239).

This question takes a slightly different form when addressed to the problem of case-based teaching, because a teacher seeks a different kind of relevance than a problem-solver. As shown in the many examples throughout this dissertation, a story that is a relevant example in a teaching situation is not always the one with the most similar index. The indexing problem as Domeshek defines it therefore does not only have to do with indices.

Ensuring access to relevant experience is not purely a problem of representing cases through indices; it also requires specifying the mechanisms for manipulating indices.[1] A good index that succinctly captures all of the relevant details in a story is an important component of retrieval, but SPIEL also needs to be able to reason about an index, for example, to determine a situation for which its associated case would be a good counter-example. In SPIEL, therefore the problem of access has two parts: the characterization of experience in the form of indices for stories (what traditionally has been considered the indexing problem), and the assessment of stories' relevance, what might be called the *relevance problem.*

The development of SPIEL has hinged on the creation of a library of explicit retrieval strategies, something that is still a rarity in case retrieval models, despite the recognition that this is an important aspect of human reasoning (Hammond 1990). Most case-based reasoning systems have been able to use a measurement of index similarity as an approximation of case utility, but the task of case-based teaching in particular seems to demand retrieval in which utility cannot be subsumed by measurement of similarity. There are many other areas in which it seems that structured notions of case utility would also be useful. Creative design, for example, requires that the designer find examples whose usefulness is partly a function of how different they are from standard solutions (Schank 1986; Wolverton and Hayes-Roth 1993). In such domains, explicit retrieval strategies will be important for building useful retrievers.

10.2.1. The relevance problem

In SPIEL, the relevance problem is a representational task: the system needs a body of knowledge to use in assessing the tutorial applicability of stories. Examples of this knowledge appear throughout the dissertation in the form of inference rules (such as manifestation and restriction rules) and special inference procedures (such as the opposite-finding algorithm).

Other case-based teaching systems, such as Creanimate, do not have knowledge bases that are explicitly employed in relevance assessment. The reason is that knowledge behind relevance decisions can be factored out of a retrieval system simply by building indices that represent conditions of relevance for cases. Indices in such a system represent not the case, but the set of situations in which the case should be retrieved. This passes the burden of understanding a case's possible uses onto the index creation process. Where case libraries are indexed manually, this approach has considerable appeal. However, it does make the indexing task itself more difficult. An indexer must interpret a case and infer its possible tutorial application in order to index it. SPIEL requires only that the indexer describe the case itself, after which the program determines the conditions of relevance. In SPIEL, the solution to the indexing problem and the relevance problem are kept separate.

---

[1]McDermott (1987) makes the argument that a representation is meaningless without algorithms for its manipulation.

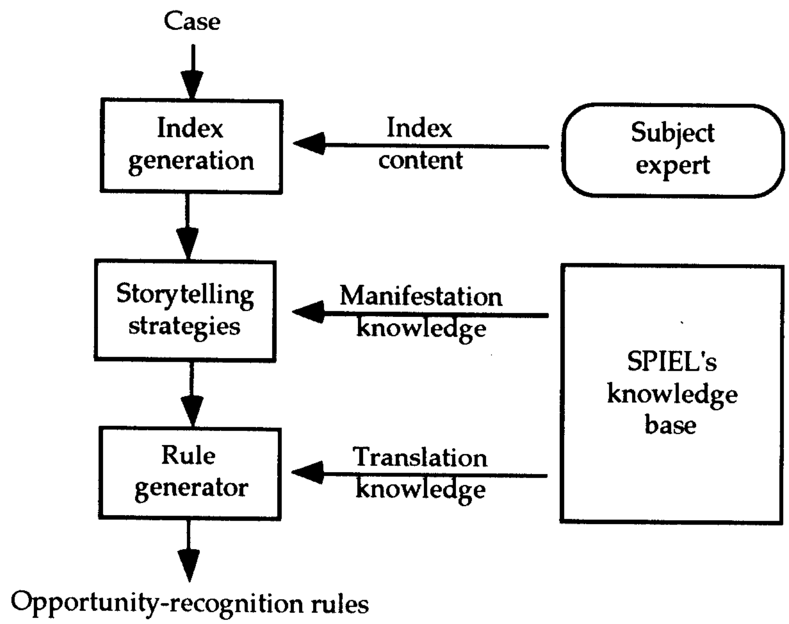

Figure 10.1. SPIEL's knowledge requirements

SPIEL's separate body of relevance knowledge makes it possible to experiment with new retrieval strategies on already-indexed stories without re-indexing them. This is a practical reason to prefer a system in which the retrieval strategies are explicit. It also makes sense from a theoretical case-based reasoning standpoint. If retrieval purposes are implicitly built into the indices used by a reasoner, it will be difficult for it to learn new case usage strategies. Explicit strategies make it possible to use old cases for new purposes without completely reindexing the contents of memory.

10.3. Future work: knowledge acquisition for strategy application

The solution to the indexing and relevance problems in SPIEL raises a new representational challenge for a retrieval system: representing the knowledge involved in relevance decisions. It should be clear from the explanation of strategy application and rule generation that forms the bulk of this dissertation that assessing relevance is a knowledge-intensive task. In SPIEL, each concept found in an index must have associated manifestation knowledge, and often several different manifestations are needed for different contexts. On top of this, SPIEL must also know the correspondences between what it seeks to recognize and the terms in which these concepts are expressed in GuSS's learning environment. Creating the knowledge base needed for SPIEL's storage phase is a laborious knowledge-engineering task.

SPIEL is knowledge-intensive because it is designed to be entirely autonomous. The user enters indices describing stories and then lets the program churn away, developing its storytelling opportunities and recognition rules. (Figure 10.1 summarizes the three knowledge types used by SPIEL in assessing relevance.) For this autonomous operation to be possible, the system must be prepared with sufficient knowledge to handle any story that comes its way, hence the large knowledge base. The Teaching Executive approach, described in chapter 2, avoids this by using user input to label important aspects of the learning environment, eliminating the need for a comprehensive translation facility between tutor and learning environment, one of the major forms of knowledge used by SPIEL. While the Teaching Executive approach does not lend itself directly to SPIEL's case retrieval problem (see the discussion in chapter 2), a similar kind of reliance on the user can be used to get around the knowledge-engineering problem in SPIEL.

A promising direction to proceed is to follow the lead of the Sounding Board program (Kass 1991). Sounding Board was a program for creative brainstorming whose knowledge was in the form of questions for elaborating explanations (Schank 1986). These questions were originally proposed as part of a mechanism for getting programs to build better explanations on their own, but in Sounding Board the questions were used to query the user about problems and possible solutions. For example, the user might tell the system "Bill never does his status reports on time." The system, knowing that people usually do things because they are motivated to do them, asks "Why do you think Bill is not doing status reports on time?" Each question contributes to the construction of an overall explanation for the problem and to the analysis of different possible solutions. Although Sounding Board cannot itself explain why Bill acts as he does, it can help a user construct an explanation by knowing how to go about the explanation process.

The use of reasoning strategies to guide a user's thinking is a powerful application of artificial intelligence ideas to the creation of useful software tools. SPIEL's storytelling strategies are good candidates for this kind of use. They can be used to guide a user's thinking about storytelling opportunities just as they are used to guide SPIEL's reasoning. For example, if SPIEL is trying to recognize that the student is failing to gather information, it has to find out what actions a student might take that would be good evidence of such a failure. Currently this knowledge comes from the system's knowledge base, but it could just as easily be gotten interactively from a user. The knowledge-intensive steps required to produce recognition condition descriptions and opportunity recognition rules need not be performed by the system autonomously.

A system that was organized in this way would not need to have complete indices for every story at the start. SPIEL currently requires that a user fully specify an index for a story so that it has enough information to apply any of its storytelling strategies. By retaining the user in the loop, the system can allow the index to be built up incrementally in the service of attempting to apply different storytelling strategies. The reader may note that I have come full circle from the initial argument in this chapter, from arguing against the co-mingling of case description and analysis of relevance within an indexing system, to the proposal of a system that does exactly that. However, the distinction between the description of a case and the assessment of its relevance is still important. It is this distinction that allows the storytelling strategies to function: to elicit information about a case and guide the user in the application of relevance knowledge.

Figure 10.2 shows a design for an interactive indexing and strategy application tool.[2] This tool builds on the framework of SPIEL, but each step (index creation, strategy application and rule generation) now requires close involvement of a user as the source of knowledge that, in SPIEL, came from the system's knowledge base.

The tool builds up its knowledge of a story and creates recognition conditions at the same time. It might ask the user "What is surprising about this story?" in order to get some notion of the kind of anomaly present and to determine what strategies to apply. Once a strategy is selected, the system can ask questions like "What plan was the salesperson using in the story?" or "What outcome would be considered opposite to the one that actually happened?" Like the strategy application process in SPIEL, the output of this process is a recognition condition description. The translation task is handled by a separate process involving a different class of user, one with expertise in the learning environment. The user answers questions that help the system translate the recognition condition descriptions into recognition procedures, avoiding the need for an exhaustive translation capacity.

---

[2]This tool is currently under development by Alex Kass, Smadar Kedar and myself.

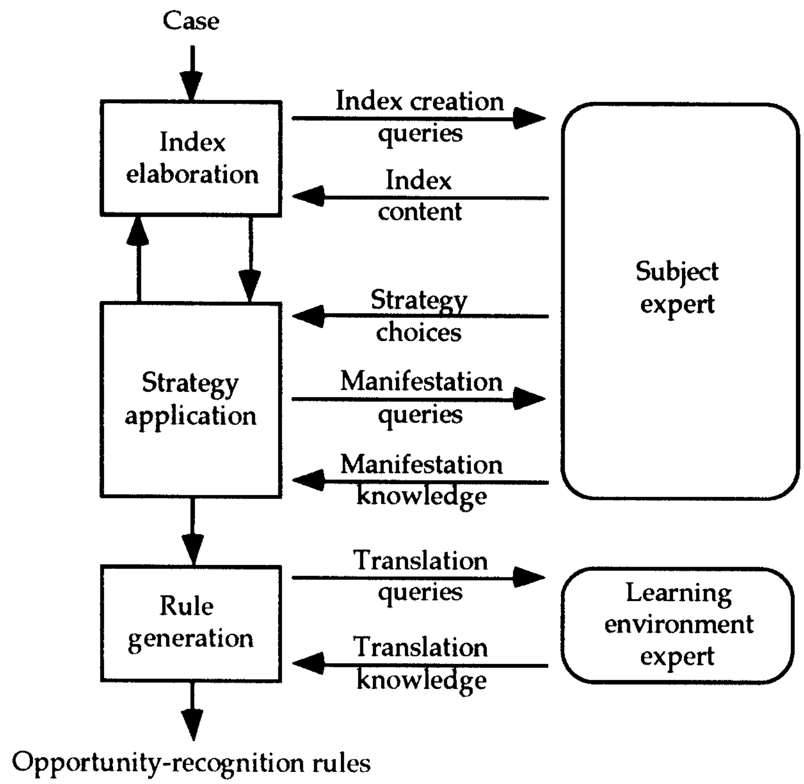

Figure 10.2. Design for a general case-based teaching tool

This tool would not only be applied in GuSS. It is a framework for augmenting any learning environment with case-based teaching. Even systems whose stories are hard-wired into place could use the tool as a methodology for placing links appropriately. What remains is for the framework to be filled in: with indexing schemes and case presentation strategies for a variety of case types and domains, and with mechanisms for creating recognition procedures for different learning environments.

Appendix A. An example rule set

This appendix contains a complete rule set for a storytelling opportunity, the one resulting from applying the **Warn about assumptions** strategy to the story "Wife watching TV." The rule specification for this rule set is shown in figure A.1. The titles given to individual rules reflect their correspondence with this diagram.

Parasitic #1: Identify non-buyer who has spouse that is business partner.

```
;; PARASITIC D-Rule <SPIEL-PARA-DRULE-1953> for story: "Wife watching TV"
;; For AGENT
;; Vars: ((*SELF #<agent>)
;;          (*AGENT1935 #<agent>)
;;          (*AGENT1936 #<agent>))
;; Indices:      ((AWARE (*SELF #<know about> #<anything> #<time>)))
;; Only-when:  ((BELIEVE (*SELF #<is spouse> #<anything> *AGENT1935))
;;                 (BELIEVE (*SELF #<is business partner> #<anything> *AGENT1935)))
;; Not-when:    ((EXPECT (*SELF #<possess> #<control of> #<business>)))
;; Peek:          <none>
;; Not-peek:     ((AWARE (SPIEL-AGENT #<linking som 1937> #<anything> *AGENT1935))
;;                  (AWARE (SPIEL-AGENT #<linking som 1936> #<anything> *SELF)))
;; Agent peeked: SPIEL-AGENT
;; Poke:          ((AWARE (SPIEL-AGENT #<linking som 1937> #<anything> *AGENT1935))
;;                 (AWARE (SPIEL-AGENT #<linking som 1936> #<anything> *SELF)))
;; Agent poked: SPIEL-AGENT
;; Action:        <none>
;; Forget:        <none>
```

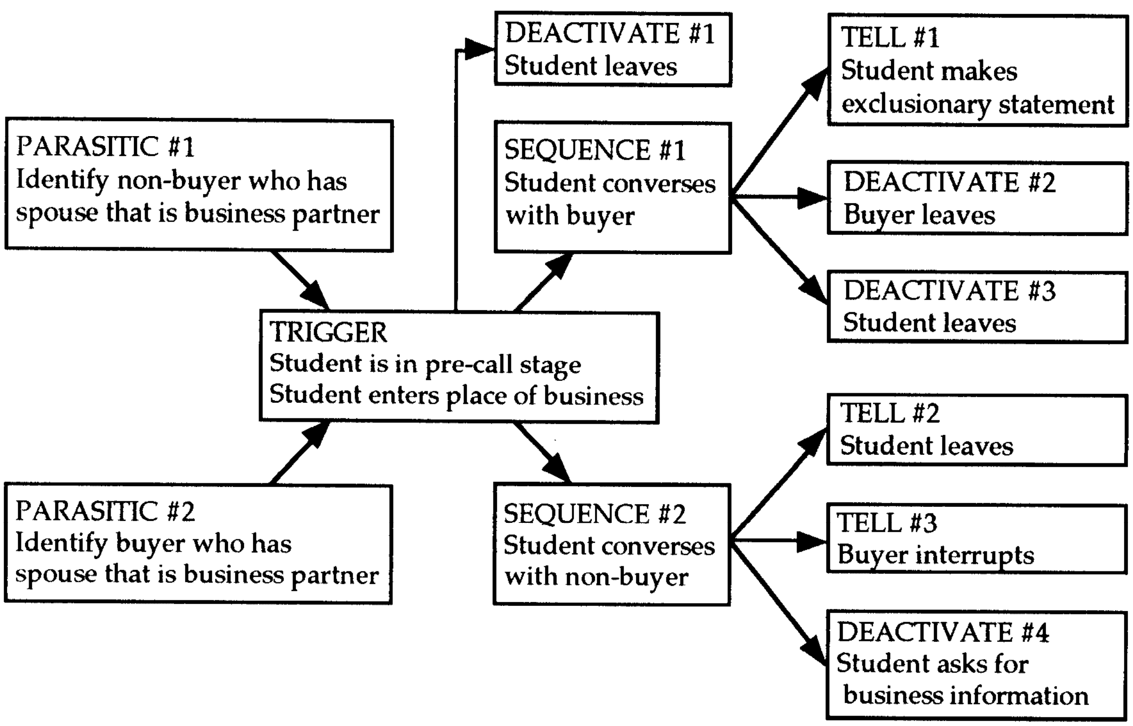

Figure A.1. Rule specification for "Wife watching TV" told using **Warn about assumptions** strategy

```
;; Learn:          <none>
```

Parasitic #2: Identify buyer who has spouse that is business partner.

```
;; PARASITIC D-Rule <SPIEL-PARA-DRULE-1952> for story: "Wife watching TV"
;; For AGENT
;; Vars: ((*SELF #<agent>)
;;         (*AGENT1935 #<agent>)
;;         (*AGENT1936 #<agent>))
;; Indices:        ((AWARE (*SELF #<know about> #<anything> #<time>)))
;; Only-when: ((BELIEVE (*SELF #<is spouse> #<anything> *AGENT1936))
;;              (BELIEVE (*SELF #<is business partner> #<anything> *AGENT1936))
;;              (EXPECT (*SELF #<possess> #<control of> #<business>)))
;; Not-when:   <none>
;; Peek:             <none>
;; Not-peek:   ((AWARE (SPIEL-AGENT #<linking som 1939> #<anything> *AGENT1936))
;;              (AWARE (SPIEL-AGENT #<linking som 1938> #<anything> *SELF)))
;; Agent peeked: SPIEL-AGENT
;; Poke:        ((AWARE (SPIEL-AGENT #<linking som 1939> #<anything> *AGENT1936))
;;              (AWARE (SPIEL-AGENT #<linking som 1938> #<anything> *SELF)))
;; Agent poked: SPIEL-AGENT
;; Action:     <none>
;; Forget:     <none>
;; Learn:      <none>
```

Trigger: Notice when student is the pre-call stage and enters the customer's place of business.

```
;; TRIGGER D-Rule <SPIEL-DRULE-1952> for story: "Wife watching TV"
;; For SPIEL-AGENT
;; Vars: ((*SELF #<SPIEL-AGENT "Storyteller">)
;;           (*LOCATION1939 #<location>)
;;           (*AGENT1935 #<agent>)
;;           (*AGENT1936 #<agent>))
;; Indices: ((AWARE (*SELF #<linking som 1936> #<anything> *AGENT1936))
;;             (AWARE (*SELF #<linking som 1937> #<anything> *AGENT1935))
;;             (AWARE (*SELF #<linking som 1939> #<anything> *AGENT1936))
;;             (AWARE (*SELF #<linking som 1938> #<anything> *AGENT1935))
;;             (AWARE (#<USER> #<is in stage> #<anything> #<Precall>))
;;             (OBSERVE (#<USER> #<enter> #<anything> *LOCATION1939)))
;; Only-when:   <none>
;; Not-when:    ((AWARE (*SELF #<linking som 1941> #<anything> *LOCATION1939)))
;; Peek:              <none>
;; Not-peek:    <none>
;; Agent peeked: <none>
;; Poke:        <none>
;; Agent poked: <none>
;; Action:      <none>
;; Forget:      <none>
;; Learn:   ((AWARE (*SELF #<linking som 1941> #<anything> *LOCATION1939)))
```

Sequence #1: Student converses with the buyer.

```
;; SEQ D-Rule <SPIEL-DRULE-1950> for story: "WWTV Test"
;; For SPIEL-AGENT
;; Vars:((*SELF #<SPIEL-AGENT "Storyteller">)
;;          (*LOCATION1939 #<location>)
;;          (*AGENT1935 #<agent>)
;;          (*AGENT1936 #<agent>))
;; Indices: ((AWARE (*SELF #<linking som 1938> #<anything> *AGENT1935))
```

```
;;              (AWARE (*SELF #<linking som 1941> #<anything> *LOCATION1939))
;;              (OBSERVE
;;           (#<USER Mrs. So AndSo> #<say something about> #<anything> #<anything>)))
;; Only-when:   ((AWARE (*AGENT1935 #<is at location> #<anything> *LOCATION1939)))
;; Not-when:    ((AWARE (*SELF #<linking som 1942> #<anything> *LOCATION1939)))
;; Peek:        <none>
;; Not-peek:    <none>
;; Agent peeked:  <none>
;; Poke:                <none>
;; Agent poked: <none>
;; Action:      <none>
;; Forget:    ((AWARE (*SELF #<linking som 1941> #<anything> *LOCATION1939)))
;; Learn:     ((AWARE (*SELF #<linking som 1942> #<anything> *LOCATION1939)))
```

Sequence #2: Student converses with the buyer's spouse.

```
;; SEQ D-Rule <SPIEL-DRULE-1944> for story: "Wife watching TV"
;; For SPIEL-AGENT
;; Vars: ((*SELF #<SPIEL-AGENT "Storyteller">)
;;          (*LOCATION1939 #<location>)
;;          (*AGENT1935 #<agent>)
;;          (*AGENT1936 #<agent>))
;; Indices:  ((AWARE (*SELF #<linking som 1939> #<anything> *AGENT1936))
;;              (AWARE (*SELF #<linking som 1941> #<anything> *LOCATION1939))
;;              (AWARE (*SELF #<linking som 1938> #<anything> *AGENT1935))
;;              (OBSERVE (#<USER> #<say something about> #<anything> #<anything>)))
;; Only-when:  ((AWARE (*AGENT1936 #<is at location> #<anything> *LOCATION1939)))
;; Not-when:  ((AWARE (*SELF #<linking som 1943> #<anything> *LOCATION1939))
;;                  (AWARE (*AGENT1935 #<is at location> #<anything> *LOCATION1939)))
;; Peek:        <none>
;; Not-peek:    <none>
;; Agent peeked:  <none>
;; Poke:        <none>
;; Agent poked: <none>
;; Action:      <none>
;; Forget:    ((AWARE (*SELF #<linking som 1941> #<anything> *LOCATION1939)))
;; Learn:     ((AWARE (*SELF #<linking som 1943> #<anything> *LOCATION1939)))
```

Deactivate #1: Student leaves without speaking to buyer or spouse.

```
;; DEACTIVATE D-Rule <SPIEL-DRULE-1943> for story: "WWTV Test"
;; For SPIEL-AGENT
;; Vars:((*SELF #<SPIEL-AGENT "Storyteller">)
;;         (*LOCATION1939 #<location>)
;;         (*AGENT1935 #<agent>)
;;         (*AGENT1936 #<agent>))
;; Indices:  ((AWARE (*SELF #<linking som 1941> #<anything> *LOCATION1939))
;;              (OBSERVE
;;           (#<USER Mrs. So AndSo> #<leave> #<anything> *LOCATION1939)))
;; Only-when:   <none>
;; Not-when:    <none>
;; Peek:        <none>
;; Not-peek:    <none>
;; Agent peeked:  <none>
;; Poke:        <none>
;; Agent poked: <none>
;; Action:      <none>
```

```
;; Forget:    ((AWARE (*SELF #<linking som 1941> #<anything> *LOCATION1939)))
;; Learn:          <none>
```

Deactivate #2: Buyer ends conversation with student.

```
;; DEACTIVATE D-Rule <SPIEL-DRULE-1948> for story: "WWTV Test"
;; For SPIEL-AGENT
;; Vars:((*SELF #<SPIEL-AGENT "Storyteller">)
;;            (*LOCATION1939 #<location>)
;;            (*AGENT1935 #<agent>)
;;            (*AGENT1936 #<agent>))
;; Indices:  ((AWARE (*SELF #<linking som 1938> #<anything> *AGENT1935))
;;                 (AWARE (*SELF #<linking som 1942> #<anything> *LOCATION1939))
;;                 (OBSERVE (*AGENT1935 #<leave> #<anything> *LOCATION1939)))
;; Only-when:    <none>
;; Not-when:     <none>
;; Peek:         <none>
;; Not-peek:     <none>
;; Agent peeked:  <none>
;; Poke:         <none>
;; Agent poked: <none>
;; Action:       <none>
;; Forget:    ((AWARE (*SELF #<linking som 1942> #<anything> *LOCATION1939)))
;; Learn:        <none>
```

Deactivate #3: Student ends conversation with buyer.

```
;; DEACTIVATE D-Rule <SPIEL-DRULE-1949> for story: "WWTV Test"
;; For SPIEL-AGENT
;; Vars:((*SELF #<SPIEL-AGENT "Storyteller">)
;;            (*LOCATION1939 #<location>)
;;            (*AGENT1935 #<agent>)
;;            (*AGENT1936 #<agent>))
;; Indices:  ((AWARE (*SELF #<linking som 1942> #<anything> *LOCATION1939))
;;                 (OBSERVE
             (#<USER Mrs. So AndSo> #<leave> #<anything> *LOCATION1939)))
;; Only-when:    <none>
;; Not-when:     <none>
;; Peek:         <none>
;; Not-peek:     <none>
;; Agent peeked:  <none>
;; Poke:         <none>
;; Agent poked: <none>
;; Action:       <none>
;; Forget:    ((AWARE (*SELF #<linking som 1942> #<anything> *LOCATION1939)))
;; Learn:        <none>
```

Deactivate #4: Student gathers business information from the spouse.

```
;; DEACTIVATE D-Rule <SPIEL-DRULE-1946> for story: "Wife watching TV"
;; For SPIEL-AGENT
;; Vars: ((*SELF #<SPIEL-AGENT "Storyteller">)
;;             (*LOCATION1939 #<location>)
;;            (*AGENT1935 #<agent>)
;;            (*AGENT1936 #<agent>))
;; Indices:   ((AWARE (*SELF #<linking som 1943> #<anything> *LOCATION1939))
;;                 (OBSERVE (#<USER> #<ask about> #<anything> #<industry term>)))
;; Only-when:    <none>
;; Not-when:     <none>
```

```
;; Peek:          <none>
;; Not-peek:      <none>
;; Agent peeked:  <none>
;; Poke:          <none>
;; Agent poked: <none>
;; Action:        <none>
;; Forget:    ((AWARE (*SELF #<linking som 1943> #<anything> *LOCATION1939)))
;; Learn:         <none>
```

Tell #1: Student makes exclusionary remark.

```
;; TELL D-Rule <SPIEL-DRULE-1951> for story: "WWTV Test"
;; For SPIEL-AGENT
;; Vars:((*SELF #<SPIEL-AGENT "Storyteller">)
;;          (*LOCATION1939 #<location>)
;;          (*AGENT1935 #<agent>)
;;          (*AGENT1936 #<agent>))
;; Indices:    ((AWARE (*SELF #<linking som 1939> #<anything> *AGENT1936))
;;          (AWARE (*SELF #<linking som 1942> #<anything> *LOCATION1939))
;;          (OBSERVE (#<USER Mrs. So AndSo> #<tell about> #<exclusion of> *AGENT1936)))
;; Only-when:     <none>
;; Not-when:      <none>
;; Peek:          <none>
;; Not-peek:      <none>
;; Agent peeked:  <none>
;; Poke:          <none>
;; Agent poked: <none>
;; Action:    ((INTERFACE (TELL "Wife watching TV" #<anything> WAS1))
;;             (INTERFACE (TELL #<headline>  #<anything> "A warning about something you just did"))
;;             (INTERFACE (TELL #<bridge>  #<anything> "If you assume that *agent1936-title*
*agent1936-last-name* will not have a role in the business, you may be surprised.  Here is a story in which an
account executive made similar assumption that did not hold:"))
;;             (INTERFACE (TELL #<coda>  #<anything> "An assumption that a spouse will not participate
in the business may be unrealistic.")))
;; Forget:    ((AWARE (*SELF #<linking som 1942> #<anything> *LOCATION1939)))
;; Learn:         <none>
```

Tell #2: Student ends conversation with spouse.

```
;; TELL D-Rule <SPIEL-DRULE-1945> for story: "WWTV Test"
;; For SPIEL-AGENT
;; Vars:((*SELF #<SPIEL-AGENT "Storyteller">)
;;          (*LOCATION1939 #<location>)
;;          (*AGENT1935 #<agent>)
;;          (*AGENT1936 #<agent>))
;; Indices:  ((AWARE (*SELF #<linking som 1943> #<anything> *LOCATION1939))
;;               (AWARE (*SELF #<linking som 1936> #<anything> *AGENT1936))
;;               (OBSERVE (#<USER Mrs. So AndSo> #<leave> #<anything> *LOCATION1939)))
;; Only-when:     <none>
;; Not-when:      <none>
;; Peek:          <none>
;; Not-peek:      <none>
;; Agent peeked:  <none>
;; Poke:          <none>
;; Agent poked: <none>
;; Action:    ((INTERFACE (TELL "Wife watching TV" #<anything> WAS1))
;;             (INTERFACE (TELL #<headline>  #<anything> "A warning about something you just did"))
;;             (INTERFACE (TELL #<bridge>  #<anything> "If you assume that *agent1936-title*
```

agent1936-last-name will not have a role in the business, you may be surprised. Here is a story in which an account executive made similar assumption that did not hold:"))
;; (INTERFACE (TELL #<coda> #<anything> "An assumption that a spouse will not participate in the business may be unrealistic.")))
;; Forget: ((AWARE (*SELF #<linking som 1943> #<anything> *LOCATION1939)))
;; Learn: <none>

Tell #3: Buyer ends student's conversation with the spouse.

;; TELL D-Rule <SPIEL-DRULE-1947> for story: "Wife watching TV"
;; For SPIEL-AGENT
;; Vars: ((*SELF #<SPIEL-AGENT "Storyteller">)
;; (*LOCATION1939 #<location>)
;; (*AGENT1935 #<agent>)
;; (*AGENT1936 #<agent>))
;; Indices: ((AWARE (*SELF #<linking som 1943> #<anything> *LOCATION1939))
;; (AWARE (*SELF #<linking som 1938> #<anything> *AGENT1935))
;; (AWARE (*SELF #<linking som 1936> #<anything> *AGENT1936))
;; (OBSERVE (*AGENT1935 #<enter> #<anything> *LOCATION1939)))
;; Only-when: <none>
;; Not-when: <none>
;; Peek: <none>
;; Not-peek: <none>
;; Agent peeked: <none>
;; Poke: <none>
;; Agent poked: <none>
;; Action: ((INTERFACE (TELL "Wife watching TV" #<anything> WAS1))
;; (INTERFACE (TELL #<headline> #<anything> "A warning about something you just did"))
;; (INTERFACE (TELL #<bridge> #<anything> "If you assume that *agent1936-title* *agent1936-last-name* will not have a role in the business, you may be surprised. Here is a story in which an account executive made similar assumption that did not hold:"))
;; (INTERFACE (TELL #<coda> #<anything> "An assumption that a spouse will not participate in the business may be unrealistic.")))
;; Forget: ((AWARE (*SELF #<linking som 1943> #<anything> *LOCATION1939)))
;; Learn: <none>

Appendix B. Representing storytelling strategies

SPIEL's storytelling strategies are defined in a simple declarative language. This appendix describes the general form of a strategy definition and the set of operations allowed in the language. It also shows the representation of each of SPIEL's 13 strategies.

B.1. The strategy definition

The basic structure of a storytelling strategy is as follows:

```
(defStrat name description
      :filter filter_conditions
      :recognition recognition_condition_constructors
      :headline headline_template
      :bridge bridge_template
      :coda coda_template
      :vars template_variable_bindings)
```

Here is the **Demonstrate risks** storytelling strategy expressed in this form:

```
(defStrat DRK
      "Tell a story about a negative result of a particular course of
action when the student has just executed a similar course of action
but had success."
      :filter
      (and (:actual-result m-negative-result)
           (:agent m-student-role))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen)
       (:actual-theme :manif)
       (:actual-goal :manif)
       (:actual-plan :manif)
       (:actual-result :result-opp :manif))
      :headline "A story showing the risks of your approach."
      :bridge "You <plan>.  Nothing bad happened but here is a story
in which doing that led to problems:"
      :coda "Nothing bad happened to you as a result of <plan>, but
sometimes <bad-result>."
      :vars ((plan :store :actual-plan)
             (bad-result :store :actual-result)))
```

The mechanism of the strategy lies in the first two parts after the description: the filter conditions, instructions for how to test an index to see if it is compatible with the strategy; and the recognition condition constructors, instructions for how to manipulate an index to arrive at a recognition condition description.

The basic operation in the filter conditions is comparison. To represent comparison, the name of a slot is paired with the name of a MOP. The condition is considered met if the filler of the slot in the index satisfies the condition of the MOP. For example, `(:actual-result m-negative-result)` would be satisfied by any index in which the actual column contained as result that was some kind of negative result.[1] These tests can be combined using the logical operators `and`, `or` and `not`.

One special filtering predicate is required by the strategies that explain the behavior of others. The `in-student-stereotype?` predicate is used to test whether MOPs correspond to the static student stereotype. Only stories that deviate from what the student is expected to know are told using these strategies. For example, in the **Explain other's plan** strategy, the following test is performed: `(in-student-stereotype? :agent :actual-goal :actual-plan)`. The predicate tests whether the stereotype students are likely to associate with characters of type `agent` includes the actual goal and actual plan from the index.

The recognition condition constructors are associations between slots and operations. To generate recognition conditions for an index, the strategy extracts the value from each slot and performs the associated operations. An example is the recognition section from the **Warn about optimism** strategy:

```
((:soc-setting :gen)
 (:phys-setting :gen :prep)
 (:anomaly :search :belief :anomaly-manif))).
```

This declaration calls on SPIEL to take three steps to create the recognition conditions: (1) it extracts the values from the social setting slot and performs generalization, eliminating elements without an explanatory connection to the index. (2) For the physical setting slot, a similar generalization is performed, followed by inference to find the setting in which the student might prepare for action such as that found in the story. (3) Then the anomaly is examined, and anomaly derivation search is performed to find the most salient anomaly to be recognized. Once the salient anomaly is located, it is converted into a belief to be recognized: the student hopes for one thing as opposed to another. Then, manifestation knowledge is applied to find observable indicators of that belief.

B.2. Inference types

There are six types of inference performed in the service of creating RCDs. They are shown in table B.1. Opposite-finding and manifestation inference each contain several sub-types. Opposite-finding computation is different for results and plans as discussed in Chapter 5. The computation of an opposite result is a simple application of opposite links from one type of goal impact to another, but finding the aspect of the outcome where the difference between columns lies may involve some search. This distinguishes this operation from the computation of opposite plans.

Under manifestation, there are four sub-types. Finding manifestations of anomalous beliefs requires identifying evidence for the anomalous belief and finding evidence against the actual contrasting belief (restrictions), so anomalies use a special version of the manifestation inference. For **Reinforce plan**, the creation of recognition conditions must include the stipulation that a plan has not already failed, to prevent the strategy from stealing storytelling opportunities from the **Demonstrate opportunities** strategy. The manifestation of plan for the purposes of reinforcing its use therefore includes the test for the plan's failure conditions, something that does not appear in ordinary reasoning about manifestations. A similar issue arises for the **Warn about plan** strategy, so there is also a special type of manifestation computation for the purposes of warning.

---

[1] For the definition for the satisfiedp predicate, see (Riesbeck and Schank 1989).

Table B.1. Inference types and labels in storytelling strategy definitions

Inference Type	Label	Description
Generalization	:gen	Generalization of setting.
Manifestation	:manif	Identification of observable manifestations
	:anomaly-manif	Identifying manifestations of an anomalous belief.
	:reinforce-manif	Finding the manifestations of a plan for the purposes of reinforcement.
	:warn-manif	Find the manifestations of a plan for the purposes of warning.
Opposite	:result-opp	Computing an opposite result.
	:plan-opp	Computing an opposite plan.
Anomaly search	:search	Searches for a salient anomaly.
Preparatory setting	:prep	Finding the setting in which preparation occurs.
Belief conversion	:belief	Turning an anomaly into a corresponding belief.
Remote indication	:remote	Finding indicators that show an outcome which would normally occur in one context will actually occur in some other context.

B.3. Natural language templates

Also associated with each strategy is a set of natural language templates for the headline, bridge, and coda, the three texts that explain each reminding to the student. The templates are English-language texts with embedded variables that are replaced by short phrases to build complete explanations. The last piece of the strategy definition is a set of variable assignments that specify what values to use to create the texts to replace the variables in the templates.

Some variables will have their values known at storage time (such as the plan that the recognition conditions attempt to identify the student using), others will only be known at retrieval time when the student takes action (such as the person to whom the student asks a certain question). In the variable assignments, each variable is associated with a descriptor indicating when that piece of the template can be filled (storage or retrieval time) and where in the index the corresponding MOP can be found. For retrieval time values, the MOP in the index is used to match against variables that are created at rule generation time, to determine which variable has the correct value.

In **Demonstrate alternative plan**, the variable assignment also makes use of the opposite-finding inference mechanism. The reason is that the bridge must refer to the plan that SPIEL tries to recognize the student following, such as the "cursory information gathering" plan in the "Long-term goals" example in Chapter 5. This plan is the result of applying the opposite-finding algorithm to the actual plan in the index, so the bridge must perform the same inference to incorporate it. The variable assignment uses a syntax similar to what is used in the recognition condition constructor: `(plan :store :actual-plan :opp)`. It calls on SPIEL to extract the actual plan at storage time, and also perform the opposite inference before creating natural language text to explain the storytelling opportunity.

B.4. Alternative-finding strategies

Demonstrate risks

```
(defStrat DRK
     "Tell a story about a negative result of a particular course of
action when the student has just executed a similar course of action but
had success."
     :filter
     (and (:actual-result m-negative-result)
          (:agent m-student-role))
     :recognition
     ((:soc-setting :gen)
      (:phys-setting :gen)
      (:actual-theme :manif)
      (:actual-goal :manif)
      (:actual-plan :manif)
      (:actual-result :result-opp :manif))
     :headline "A story showing the risks of your approach."
     :bridge "You <plan>.  Nothing bad happened but here is a story in
which doing that led to problems:"
     :coda "Nothing bad happened to you as a result of <plan>, but
sometimes <bad-result>."
     :vars ((plan :store :actual-plan)
            (bad-result :store :actual-result)))
```

Demonstrate opportunities

```
(defStrat DOP
     "Tell a story about a positive result of a particular course of
action when the student has just executed a similar course of action but
had a poor result."
     :filter
     (and (:actual-result m-positive-result)
          (:agent m-student-role))
     :recognition
     ((:soc-setting :gen)
      (:phys-setting :gen)
      (:actual-theme :manif)
      (:actual-goal :manif)
      (:actual-plan :manif)
      (:actual-result :result-opp :manif))
     :headline "A story that shows an opportunity that didn't arise
here."
     :bridge "You <plan>.  It didn't work well, but here is a story in
which that strategy worked:"
     :coda "You didn't have a good result from <plan>, but sometimes
<good-result>."
     :vars ((plan :store :actual-plan)
            (good-result :store :actual-result)))
```

Demonstrate alternative plan

```
(defStrat DAP
     "Tell a story about a successful plan to achieve a particular goal
when the student has executed a different plan and failed to achieve the
goal."
     :filter
```

```
      (and (:actual-result m-positive-result)
           (:agent m-student-role))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen)
       (:actual-theme :manif)
       (:actual-goal :manif)
       (:actual-plan :plan-opp :manif)
       (:actual-result :result-opp :manif))
      :headline "A story about a different approach you might try"
      :bridge "You <plan> without much success.  Here is a story about a
similar situation in which <gen-actor> used a different method and was
successful."
      :coda "You <plan>.  It didn't work well.  In the future, you might
consider <story-plan>."
      :vars ((plan :store :actual-plan :opp)
             (story-plan :store :actual-plan)
             (gen-actor :store :agent)))
```

B.5. Critiquing strategies

Warn about optimism, student focused

```
(defStrat WOP1
      "Tell a story about a desire that a salesperson had about their own
actions which was not realized when the student appears have a similar
desire. "
      :filter
      (and (:view-result m-positive-result)
           (not (:actual-result m-positive-result))
           (:agent m-student-role)
           (:viewer m-student-role)
           (:perspective m-wanted))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen :prep)
       (:anomaly :search :belief :anomaly-manif))
      :headline "A warning about something you just did"
      :bridge  "If you are hoping that you <desired>, you may be
disappointed.  Here is a story in which <actor> had a similar hope that
was not realized:"
      :coda "The hope that you <desired> may be unrealistic."
      :vars ((actor :store :agent)
             (desired :store :anomaly)))
```

Warn about optimism, other focused

```
(defStrat WOP2
      "Tell a story about a desire that a salesperson had about someone
else which was not realized when the student appears have a similar
desire. "
      :filter
      (and (:view-side+ m-positive-result)
           (not (:actual-side+ m-negative-result))
           (:viewer m-student-role)
           (not (:agent m-student-role))
           (:perspective m-wanted))
```

```
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen :prep)
       (:anomaly :search :belief :anomaly-manif))
      :headline "A warning about something you just did"
      :bridge  "If you are hoping that <actor> <desired>, you may be
disappointed.  Here is a story in which <viewer> had a similar hope that
was not realized:"
      :coda "The hope that <gen-actor> <desired> may be unrealistic."
      :vars ((actor :retrieve :agent)
             (gen-actor :store :agent)
             (viewer :store :viewer)
             (desired :store :anomaly)))
```

Warn about pessimism, student focused

```
(defStrat WPS1
      "Tell a story about a fear that was not realized when the student
appears to have the same fear about themselves."
      :filter
      (and (:view-result m-positive-result)
           (not (:actual-result m-negative-result))
           (:agent m-student-role)
           (:viewer m-student-role)
           (:perspective m-feared))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen :prep)
       (:anomaly :search :belief :anomaly-manif))
      :headline "A hint about something you just did"
      :bridge "If you fear that you <fear>, you may be surprised.  Here is
a story in which <actor> had similar fears that were not realized:"
      :coda "A fear that you <fear> may be unrealistic."
      :vars ((actor :store :agent)
             (fear :store :anomaly)))
```

Warn about pessimism, other focused

```
(defStrat WPS2
      "Tell a story about a fear that was not realized when the student
appears to have the same fear about someone else."
      :filter
      (and (:view-side- m-negative-result)
           (not (:actual-side- m-negative-result))
           (:viewer m-student-role)
           (not (:agent m-student-role))
           (:perspective m-feared))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen :prep)
       (:anomaly :search :belief :anomaly-manif))
      :headline "A hint about something you just did"
      :bridge "If you fear that <actor> <fear>, you may be surprised.
Here is a story in which <viewer> had similar fears that were not
realized:"
      :coda "A fear that <gen-actor> <fear> may be unrealistic."
      :vars ((actor :retrieve :agent)
```

```
            (gen-actor :store :agent)
            (viewer :store :viewer)
            (fear :store :anomaly)))
```

Warn about assumption, student focused

```
(defStrat WAS1
      "Tell a story about an assumption that someone made which did not
hold when the student appears to have a similar assumption about
themselves."
      :filter
      (and (:agent m-student-role)
           (:viewer m-student-role)
           (or (:perspective m-assumed)
               (:perspective m-standard)))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen :prep)
       (:anomaly :search :belief :anomaly-manif))
      :headline "A warning about something you just did"
      :bridge "If you assume that you <assumption>, you may be surprised.
Here is a story in which <actor> had a similar asssumption that did not
hold:"
      :coda "An assumption that you <assumption> may be unrealistic."
      :vars ((actor :store :agent)
             (assumption :store :anomaly)))
```

Warn about assumption, other focused

```
(defStrat WAS2
      "Tell a story about an assumption that someone made which did not
hold when the student appears to have a similar assumption about someone
else."
      :filter
      (and (:viewer m-student-role)
           (not (:agent m-student-role))
           (or (:perspective m-assumed)
               (:perspective m-standard)))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen :prep)
       (:anomaly :search :belief :anomaly-manif))
      :headline "A warning about something you just did"
      :bridge "If you assume that <actor> <assumption>, you may be
surprised.  Here is a story in which <viewer> had a similar asssumption
that did not hold:"
      :coda "An assumption that <gen-actor> <assumption> may be
unrealistic."
      :vars ((actor :retrieve :agent)
             (gen-actor :store :agent)
             (viewer :store :viewer)
             (assumption :store :anomaly)))
```

B.6. Projection strategies

Reinforce plan

```
(defStrat RIP
     "Tell a story about a successful plan to achieve a particular goal
when the student has just started to execute a similar plan."
     :filter
     (and (:actual-result m-positive-result)
          (:agent m-student-role))
     :recognition
     ((:soc-setting :gen)
      (:phys-setting :gen)
      (:actual-theme :manif)
      (:actual-goal :manif)
      (:actual-plan :reinforce-manif)
      (:actual-result :remote))
     :headline  "A success story about a situation like yours."
     :bridge "<plan> is sometimes important for <goal>.  Here is a story
about how a <actor> succeeded using a similar approach."
     :coda "Sometimes <plan> is the right thing to do.  Keep working at
it."
     :vars ((actor :store :agent)
            (plan :store :actual-plan)
            (goal :store :actual-goal)))
```

Warn about plan

```
(defStrat WAP
     "Tell a story about an unsuccessful plan when the student has begun
executing a similar plan. "
     :filter
     (and (:actual-result m-negative-result)
          (:agent m-student-role))
     :recognition
     ((:soc-setting :gen)
      (:phys-setting :gen)
      (:actual-theme :manif)
      (:actual-goal :manif)
      (:actual-plan :warn-manif)
      (:actual-result :remote))
     :headline "A story about a failure in a situation similar to yours."
     :bridge "You <plan>.  Here is a story in which doing that led to
problems."
     :coda "You <plan>.  That might not be a good idea."
     :vars ((plan :store :actual-plan)))
```

B.7. Other-directed strategies

Explain other's plan

```
(defStrat XOP
     "Tell a story about a plan that the student might not know about
when the student has just observed some agent execute a similar plan."
     :filter
     (and (not (:agent m-student-role))
          (not
```

```
            (in-student-stereotype? :agent :actual-goal :actual-plan)))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen)
       (:actual-theme :manif)
       (:actual-goal :manif)
       (:actual-plan :w-manif))
      :headline "Do you understand what <actor> is doing?"
      :bridge "Sometimes <gen-actor> will <plan>.  Here is a story about a
time when that took place."
      :coda "<actor> may be trying <goal> by <plan>.  If you don't know,
you should try to find out."
      :vars ((actor :retrieve :agent)
             (gen-actor :store :agent)
             (plan :store :actual-plan)
             (goal :store :actual-goal)))
```

Explain other's perspective

```
(defStrat XOV
      "Tell a story about an expectation that the student might not
understand when the student has just observed some agent act on the basis
of a similar expectation."
      :filter
      (and (not (:agent m-student-role))
           (not (in-student-stereotype? :agent :anomaly)))
      :recognition
      ((:soc-setting :gen)
       (:phys-setting :gen)
       (:anomaly :search :belief :anomaly-manif))
      :headline "Do you understand <viewer>'s perspective in this
situation?"
      :bridge "Sometimes <gen-viewer> will <persp> that <actor>
<expectation>.  Here is a story about a time when that took place."
      :coda "<viewer> may <persp> that <actor> <expectation>.  If you
don't know, you should try to find out."
      :vars ((viewer :retrieve :viewer)
             (gen-viewer :store :viewer)
             (actor :retrieval :actor)
             (expectation :store :anomaly)))
```

References

Anderson, J. R. 1985. Cognitive Psychology and Its Implications. New York: W. H. Freeman and Co.

________. 1988. The Expert Module. In Foundations of Intelligent Tutoring Systems ed. M. C. Polson and J. J. Richardson, 21-53. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., C. F. Boyle and G. Yost. 1985. The geometry tutor. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1-7. Los Altos, CA: Morgan Kaufmann.

Ashley, K. D. 1990. Modeling Legal Argument: Reasoning with Cases and Hypotheticals. Cambridge: MIT Press.

Ashley, K. D., and V. Aleven. 1991. A Computational Approach to Explaining Case-Based Concepts of Relevance in a Tutorial Context. In Proceedings: Case-Based Reasoning Workshop II (DARPA), 257-268. San Mateo, CA: Morgan Kaufmann.

Ashley, K., and E. Rissland. 1987. Compare and Contrast, A Test of Expertise. In Proceedings of Sixth National Conference on Artificial Intelligence, 273-278. Menlo Park, CA: AAAI Press.

Bareiss, R., and J. King. 1989. Similarity Assessment in Case-Based Reasoning. In Proceedings: Case-Based Reasoning Workshop (DARPA), 67-71. San Mateo, CA: Morgan Kaufmann.

Birnbaum, L. A. 1986. Integrated Processing in Planning and Understanding. PhD Thesis, Yale University. Technical Report 489

Birnbaum, L. and G. Collins. 1989. Remindings and engineering design themes: a case study in indexing vocabulary. In Proceedings: Case-based Reasoning Workshop (DARPA), 47-51. San Mateo, CA: Morgan Kaufmann.

Blevis, E., R. Burke, G. Downey, A. Kass, C. Lewis, M. Saunders and J. Sierant. (in preparation). The Play's the thing: Authoring and content tools for the Mikrocosmos.

Blevis, E. and A. M. Kass. 1991. Teaching by Means of Social Simulation. In International Conference on the Learning Sciences: Proceedings of the 1991 Conference, 45-51. Charlottesville, VA: AACE.

Boose, J. H. 1986. Expertise Transfer for Expert System Design. New York: Elsevier.

Brownston, L., R. Farrell, E. Kant and N. Martin. 1985. Programming expert systems in OPS5. Reading, MA: Addison-Wesley.

Bruner, J. 1986. Actual Minds, Possible Worlds. Cambridge: Harvard University Press.

Burke, R. 1989. Understanding and Responding in Conversation: Case Retrieval with Natural Language. In Proceedings: Case-Based Reasoning Workshop (DARPA), 230-234. San Mateo, CA: Morgan Kaufmann.

Burns, H. L. and C. G. Capps. 1988. Foundations of Intelligent Tutoring Systems: An Introduction. In Foundations of Intelligent Tutoring Systems ed. M. C. Polson and J. J. Richardson, 1-19. Hillsdale, NJ: Lawrence Erlbaum Associates.

Burton, R. R. 1982. Diagnosing bugs in a simple procedural skill. In Intelligent Tutoring Systems ed. D. Sleeman and J. S. Brown, 157-183. New York: Academic Press.

Carbonell, J.G. 1983. Learning by Analogy: Formulating and Generalizing Plans from Past Experience. In Machine Learning: An Artificial Intelligence Approach ed. R. S. Michalski, J. G. Carbonell and T. M. Mitchell. Los Altos, CA: Morgan Kaufman.

Clancey, W. J. 1982. Tutoring rules for guiding a case method dialogue. In Intelligent Tutoring Systems ed. D. Sleeman and J. S. Brown, 201-225. New York: Academic Press.

Clancey, W. J. 1991. The Knowledge Level Reinterpreted: Modeling Socio-Technical Systems. In AAAI Spring Symposium on Cognitive Aspects of Knowledge Acquisition, 47-56. Stanford University: AAAI.

Collins, A., J. S. Brown and S. E. Newman. 1989. Cognitive Apprenticeship: Teaching the Crafts of Reading Writing, and Mathematics. In Knowing, learning, and instruction: Essays in honor of Robert Glaser ed. L. B. Resnick. Hillsdale, NJ: Lawrence Erlbaum Associates.

Craig, R. L., and L. Kelly ed. 1990. Sales training handbook: a guide to developing sales performance. Englewood Cliffs, NJ: Prentice Hall.

Domeshek, E. A. 1992. Do the Right Thing: A Component Theory for Indexing Stories as Social Advice. PhD thesis, Yale University. Issued as Technical Report #26, Institute for the Learning Sciences, Northwestern University.

Edelson, D. C. 1993. Learning from Stories: Indexing, Reminding and Questioning in a Case-based Teaching System. PhD Thesis, Northwestern University. Issued as Technical Report #43, Institute for the Learning Sciences, Northwestern University.

Elsom-Cook, M. ed. 1990. Guided Discovery Tutoring. London: Paul Chapman Publishing.

Ferguson, W., R. Bareiss, L. Birnbaum and R. Osgood. 1992. ASK Systems: An Approach to the Realization of Story-based Teachers. Institute for the Learning Sciences, Northwestern University. Technical Report #22.

Funke, J. 1991. Solving Complex Problems: Exploration and Control of Complex Systems. In Complex Problem Solving: Principles and Mechanisms ed. R. J. Sternberg and P. A. Frensch, 185-222. Hillsdale, NJ: Lawrence Erlbaum Associates.

Goldstein, I. P. 1982. The genetic graph: a representation for the evolution of procedural knowledge. In Intelligent Tutoring Systems, D. Sleeman & J. S. Brown ed., 51-77. New York: Academic Press.

Hammond, K. J. 1986. Case-based Planning: An Integrated Theory of Planning, Learning and Memory. PhD thesis, Yale University. Technical Report 488.

________. 1989. Opportunistic Memory. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 504-510. Menlo Park, CA: AAAI Press.

________. 1990. Explaining and Repairing Plans that Fail. Artificial Intelligence, 45: 173-228.

Hollan, J. D., E. L. Hutchins, and L. Weitzman. 1984. STEAMER: An interactive inspectable simulation-based training system. AI Magazine, 5(2): 15-27.

Hunter, L. 1989. Knowledge Acquisition Planning: Gaining Expertise Through Experience. PhD Thesis, Yale University. Technical Report 678.

Jona, M. Y. (in preparation). Representing and applying teaching knowledge in computer-based learning environments. PhD Thesis, Northwestern University.

Kass, A. M. 1991. Question Asking, Artificial Intelligence, and Human Creativity. Institute for the Learning Sciences, Northwestern University. Technical Report #11.

Kass, A. M. and E. Blevis. 1991. Learning Through Experience: An Intelligent Learning-by-Doing Environment for Business Consultants. In Proceedings of the Intelligent Computer-Aided Training Conference. NASA.

Kass, A. M., R. Burke, E. Blevis and M. Williamson. 1992. The GuSS Project: Integrating Instruction and Practice through Guided Social Simulation. Institute for the Learning Sciences, Northwestern University. Technical Report #34.

Kass, A. M., R. Burke, E. Blevis, and M. Williamson. (to appear). Constructing learning environments for complex social skills. Journal of the Learning Sciences.

Kass, A. M., D. B. Leake and C. C. Owens. 1986. SWALE: A Program that Explains. In Explanation Patterns: Understanding Mechanically and Creatively Roger C. Schank, 232-254. Hillsdale, NJ: Lawrence Erlbaum Associates.

Klahr, D., K. Dunbar and A. L. Fay. 1990. Designing good experiments to test bad hypotheses. In Computational models of scientific discovery ed. J. Shrager and P. Langley, 355-402. Palo Alto, CA: Morgan Kaufmann Publishers.

Kolodner, J. L. 1984. Retrieval and Organizational Strategies in Conceptual Memory. Hillsdale, NJ: Lawrence Erlbaum Associates.

________. 1989. Judging Which is the 'Best' Case for a Case-based Reasoner. In Proceedings: Case-Based Reasoning Workshop (DARPA), 77-81. San Mateo, CA: Morgan Kaufmann.

________. 1993. Case-based reasoning. San Mateo, CA: Morgan Kaufmann.

Kolodner, J. L., and L. M. Wills. 1993. Paying Attention to the Right Thing: Issues of Focus in Case-Based Creative Design. In AAAI Workshop on Case-Based Reasoning, ed. D. Leake, 19-25. Menlo Park, CA: AAAI Press.

Labov, W. 1972. The Transformation of Experience in Narrative Syntax. Chapter 9 in Language in the Inner City: Studies in the Black English Vernacular. Philadelphia: University of Pennsylvania Press.

Lave, J., and E. Wenger. 1991. Situated Learning: legitimate peripheral participation. New York: Cambridge University Press.

Leake, D. B. 1992. Evaluating Explanations: A Content Theory. Hillsdale, NJ: Lawrence Erlbaum Associates.

Leake, D. B. ed. 1993. Case-Based Reasoning: Papers from the 1993 Workshop. AAAI Press. Technical Report No. WS-93-01.

Lemke, A. C., and G. Fischer. 1990. A Cooperative Problem Solving System for User Interface Design. In Proceedings of the Eighth Annual Conference on Artificial Intelligence, 479-484. Menlo Park, CA: AAAI Press.

Lesgold, A., and S. Lajoie. 1991. Complex Problem Solving in Electronics. In Complex Problem Solving: Principles and Mechanisms ed. R. J. Sternberg and P. A. Frensch, 287-316. Hillsdale, NJ: Lawrence Erlbaum Associates.

Lesgold, A., S. Lajoie, M. Bunzo and G. Eggan. 1992. SHERLOCK: A Coached Practice Environment for an Electronics Troubleshooting Job. In Computer-Assisted Instruction and Intelligent Tutoring Systems ed. J. H. Larkin & R. W. Chabay, 201-238. Hillsdale, NJ: Lawrence Erlbaum Associates.

Martin, C. E. 1991. Direct Memory Access Parsing. PhD Thesis, Yale University.

Martin, C. E., and C. K. Riesbeck. 1986. Uniform Parsing and Inferencing for Learning. In Proceedings of the Fifth National Conference on Artificial Intelligence, 257-261. Menlo Park, CA: AAAI Press.

McDermott, D. V. 1987. A Critique of Pure Reason. Computational Intelligence, 3:151-160.

McDougal, T., K. Hammond, & C. Seifert. 1991. A Functional Perspective on Reminding. In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society, 510-515. Hillsdale, NJ: Lawrence Erlbaum.

Minsky, M. 1986. The Society of Mind. New York: Simon and Schuster.

Newman, D., P. Griffin, and M. Cole. 1989. The construction zone: working for cognitive change in school. New York: Cambridge University Press.

Ohmaye, E. 1992. Role Playing and Social Simulation. PhD Thesis, Northwestern University. Issued as Technical Report #30, Institute for the Learning Sciences, Northwestern University.

Orr, J. 1986. Narratives at work: Story telling as cooperative diagnostic activity. In Proceedings of the Conference on Computer-Supported Cooperative Work. Austin, TX.

Osgood, R., and R. Bareiss. 1993. Index Generation in the Construction of Large-scale Conversation Hypermedia Systems. In AAAI Symposium on Case-Based Reasoning and Information Retrieval, 78-89. Stanford University, AAAI.

Owens, C. C. 1990. Indexing and Retrieving Abstract Planning Knowledge. PhD Thesis, Yale University.

Papert, S. 1980. Mindstorms: children, computers and powerful ideas. New York: Basic Books.

Polanyi, L. 1985. Telling the American Story: A Structural and Cultural Analysis of Conversational Storytelling. New York: Ablex. Reprinted by MIT Press, 1990.

Polson, M. C., and J. J. Richardson ed. 1988. Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates.

Porter, B. C., R. Bareiss and R. C. Holte. 1990. Concept Learning and Heuristic Classification in Weak-Theory Domains. Artificial Intelligence, 45: 229-263.

Ram, A. 1989. Question-driven understanding: An integrated theory of story understanding, memory and learning. Phd Thesis, Yale University. Technical Report 710.

Reiser, B. V., J. R. Anderson and R. G. Farrell. 1985. Dynamic student modeling in an intelligent tutor for lisp programming. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 8-14. Los Altos, CA: Morgan Kaufmann.

Rich, E. 1986. Users are Individuals: Individualizing User Models. In Intelligent Information Systems: Progress and Prospects, ed. R. Davies. Chichester, UK: Ellis Horwood.

Riesbeck, C. K. 1993. The Future of Case-Based Reasoning. Invited talk given at the AAAI Workshop on Case-Based Reasoning, Washington, D.C.

Riesbeck, C. K., and R. C. Schank. 1989. Inside Case-Based Reasoning. Hillsdale, NJ: Lawrence Erlbaum.

Rogoff, B., and J. Lave ed. 1984. Everyday cognition: its development in social context. Cambridge: Harvard University Press.

Salton, G., and M. J. McGill. 1983. Introduction to modern information retrieval. New York: McGraw-Hill.

Schank, R. C. 1982. Dynamic Memory: A Theory of Learning in Computers and People. New York: Cambridge University Press.

________. Explanation Patterns: Understanding Mechanically and Creatively. Hillsdale, NJ: Lawrence Erlbaum Associates.

________. 1990a. Teaching Architectures. Institute for the Learning Sciences, Northwestern University. Technical Report #3.

________. 1990b. Tell Me a Story: A New Look at Real and Artificial Memory. New York: Charles Scribner's Sons.

Schank, R. C., and R. Abelson. 1977. Scripts, Plans, Goals and Understanding. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R. C., and M. Y. Jona. 1991. Empowering the Student: New Perspectives on the Design of Teaching Systems. Journal of the Learning Sciences, 1(1): 7-35.

Schank, R. C., R. Osgood, M. Brand, R. Burke, E. Domeshek, D. Edelson, W. Ferguson, M. Freed, M. Jona, B. Krulwich, E. Ohmaye and L. Pryor. 1990. A Content Theory of Memory Indexing. Institute for the Learning Sciences, Northwestern University. Technical Report #2.

Schauble, L., R. Glaser, K. Raghavan and M. Reiner. 1991. Causal models and experimentation strategies in scientific reasoning. Journal of the Learning Sciences, 1(2): 201-238.

Seifert, C. M., G. McKoon, R. P. Abelson and R. Ratcliff. 1986. Memory Connections Between Thematically Similar Episodes. Journal of Experimental Psychology: Learning, Memory and Cognition, 12(2): 220-231.

Shafto, E., R. Bariess and L. Birnbaum. 1992. A Memory Architecture for Case-Based Argumentation. In Fourteenth Annual Conference of the Cognitive Science Society, 307-312. Hillsdale, NJ: Lawrence Erlbaum Associates.

Slator, B. M., C. K. Riesbeck, K. C. Fidel, M. Zabloudil, A. Gordon, M. S. Engber, T. Offer-Yehoshua and I. Underwood. 1991. TAXOPS: Giving expert advice to experts. Institute for the Learning Sciences, Northwestern University. Technical Report #19.

Sleeman, D., and J. S. Brown ed. 1982. Intelligent Tutoring Systems. New York: Academic Press.

Stevens, S. M. 1989. Intelligent Interactive Video Simulation of a Code Inspection. Communications of the ACM, 32(7), 832-843.

vanLehn, K. 1988. Student modeling. In Foundations of intelligent tutoring systems ed. M. C. Polson & J. J. Richardson, 55-78. Hillsdale, NJ: Lawrence Erlbaum Associates.

Wilensky, R. 1983. Planning and Understanding. Reading, Mass: Addison-Wesley.

Williams, S. M. 1991. Putting Case-Based Instruction into Context: Examples from Legal, Business, and Medical Education. Technical Report, Learning Technology Center, Vanderbilt University. Reprinted in Journal of the Learning Sciences, 2(4): 367-427.

Witherell, C., and N. Noddings ed. 1991. Stories lives tell: narrative and dialogue in education. New York: Teachers College Press.

Wolverton, M., and B. Hayes-Roth. 1993. Using Controlled Knowledge Search to Retrieve Cross-Contextual Information. In AAAI Spring Symposium on Case-Based Reasoning and Information Retrieval, 133-139. Stanford University, AAAI.