1

Storytelling and Creativity

1.1 Introduction

During my senior year in college, I was browsing the stacks in the research library and quite by accident came across a small but intriguing book called *The Morphology of the Folktale* (Propp, 1968). The author, Vladimir Propp, had studied common Russian folktales and distilled the form\(^1\) of those tales into cryptic equations:

\[
S \rightarrow ABC \uparrow DEFG \frac{HJK \downarrow Pr - Rs^L}{LMJNK \downarrow Pr - Rs} QExTUW *
\]

Using letters to represent story elements such as "One of the members of a family absents himself from home," each equation captured a common pattern Propp had found in the folktales he studied. Altogether the equations in Propp’s book formed a definitive description of the form of Russian folktales.

As a computer scientist, I found this fascinating. Propp had reduced the folktale to a grammar—a set of well-defined rules. Grammars are used throughout computer science to formalize structure and to understand well-structured input. Compilers, for example, use grammars to translate computer programs in languages like FORTRAN—which are easily understood by humans—into the binary “ones and zeros” understood by computers. In theory, Propp’s grammar

\(^{1}\)Morphology is the study of form and structure.
could be programmed into a computer and used to recognize folktales—provided someone first translated each folktale into Propp’s notation.

But what was more intriguing to me was the notion of running Propp’s grammar “backwards.” Propp’s grammar was intended as a tool for recognizing and understanding the underlying forms of folktales. But, I reasoned, the same grammar could be used in reverse to create folktales. By starting with an initial rule and then randomly choosing the next rule to apply, Propp’s grammar could be used to “grow” a story from seed to completion. The random choices would ensure that the story created would likely be different from any actual folktale, while the rules would ensure that the resulting story had the form of a folktale. And programming a computer to do this would be trivial. A few hours in the Computer Lab and I would have a computer program that could tell stories!

Or so I thought.

I did eventually write a computer program that tells stories. But it took years, not hours, and in the end, Vladimir Propp’s intriguing little grammar was nowhere to be seen.

This volume is the story of the program I wrote and what I learned in the process. It looks at the myriad problems an author faces when he sits down to write a story, and presents the processes, techniques, and knowledge needed to address these problems. Like all authors, I hope you find my story both interesting and enlightening.

1.2 The Storytelling Problem

It is surprisingly difficult to tell a story. Even young children can understand stories. By four or five, children understand most aspects of folktales like the ones Propp studied. Indeed, the primary use of folktales is to teach the young principles they’ll need as adults.

But telling a story is a different matter. It seems an easy enough task. Surely an adult should be able to easily create what a child can easily understand. But more than a few educated, intelligent adults have learned differently when put on the spot by their children. Most can manage little more than an embarrassing hodge-podge of stereotypical clichés, inevitably starting “Once upon a time...” They sputter out a trite beginning and are soon lost. And if grown adults find it difficult to tell a story, imagine how much more difficult it must be to build a computer program to tell stories.

Why is storytelling so difficult? Storytelling appears simple because at a surface level, stories are simple. As Propp showed, the form of a story can be captured by a simple, easily understood formalism. But there is more to a story than form. Underlying the form is the story content—the meaning of the story. And it is here that the difficulties arise.

Because in this case, form does not reflect function. Underlying the form of a story is a complex web of author goals, reader expectations, and cultural knowledge. Just as an elegant mathematical proof does not reveal the knowledge and effort that went into its making, neither does the form of a story reflect the difficult process of its creation.

Authors craft stories to achieve a wide variety of complex and often competing goals. To understand why storytelling is so difficult, we must understand what an author is trying to achieve. To build a computer program to tell stories, we must understand and model the processes an author uses to achieve his goals. Both of these are difficult tasks.

The next few sections of this chapter illuminate some of the problems an author faces in telling a story. In the second part of this chapter we’ll take a quick look at how these problems can be solved by a computer program.

1.2.1 Why Form Alone Is Insufficient

Around 1958, Roger Price and Leonard Stern came out with a party game called Mad Libs® (Price & Stern, 1958) that became an instant classic. Each Mad Lib was a story with key words missing:

A Fable

Once upon a time there was a very curious girl who was always sticking her nose into everybody’s ______ (plural noun). She kept company with a/an ______ (adjective) man named Dave, who was always buying her ______ (adjective) presents...

The game is played by having people fill in the blanks knowing the proper type of word, but not the surrounding context. The result is often funny and occasionally hilarious.

Like Propp’s work, each Mad Lib is a kind of grammar. It specifies the form of a story without specifying the exact content of the story. Mad Libs works as a party game because the final story has a legitimate form combined with an absurd meaning. That’s a combination that is, at least in small doses, quite amusing.

But as a way to create stories, Mad Libs leave much to be desired. Mad Libs are amusing, but they aren’t good stories. Propp’s grammar, although more complex, has the same failing. It captures the form of a story but not the content of a story. And like Mad Libs, Propp’s grammar can produce stories that have good form but absurd meanings. The fundamental failing of story grammars is that they capture form without meaning.

Of course, the form of a story is important. We can appreciate a well-crafted story, or admire a good turn of phrase. We also expect a story to have a certain form, and may classify it as a “bad” story if it does not. But most of our appreciation of stories comes from the content level. Storytelling is an act of
communication between the author and his or her readers. It is what the story tells—the message—that matters most to both the author and the readers. What Mad Libs and Propp’s grammar fail to capture is the message level of storytelling. Any storytelling system based solely on the surface features of stories—whether a complex system like Propp’s or a simple system like Mad Libs—will inevitably fail to be successful. A story that doesn’t mean anything is not a story, even if it has the proper form.

A storyteller must have an in-depth understanding of the stories he or she tells.

An author must understand the meaning of the stories he or she tells. One reason storytelling is difficult is because it requires the storyteller to understand the story at every level: the surface format, the message or point of the story, the actions of the characters, the events in the story world, the literary values in the story.

For human authors, this task isn’t difficult. Humans spend the first 20 years of their lives learning about the world, about how people act, and about ways to understand the world. They are skilled and experienced at using this knowledge, whether to manage their day-to-day life or to understand a story.

But for a computer program this represents a tremendous barrier. To a certain extent a story is a model of a tiny world, peopled with story characters, natural forces, locations, and inanimate objects. To understand these things and their interrelationships requires a tremendous amount of knowledge that humans take for granted. Consider, for example, the simple sentence:

When Galahad saw the dragon charge, he drew his sword and jumped to the side.

To understand this sentence in depth requires an enormous amount of knowledge:

- What is a dragon? What is a knight?
- What does “charging” mean in this context? “Drawing?”
- What is the dragon trying to accomplish? Galahad?
- What will Galahad do next? The dragon?
- What is Galahad feeling? The dragon?

Capturing and applying all this knowledge to the task of storytelling is one of the challenges of building a computer program that can tell stories. But this is necessary because a story is more than just a form; it has in-depth meaning to both the author and the reader.

1.2.2 Purpose and Message

Meaning is important to storytelling because storytelling is a form of communication. The author of a story isn’t simply stringing together words randomly, or even according to a grammar. Storytelling is a purposeful activity. The storyteller constructs his story to bear a message to his or her readers.

Consider the following story:

**Rainy Day**

One day, Tom got up in the morning and saw that it was raining. He went downstairs and had breakfast. Then he sat by the window and read a book for a while. It was still raining. Later, Tom fixed himself a sandwich for lunch.

Rainy Day isn’t much of a story. The problem isn’t that it lacks form (the sentences are all grammatical) or that it has an absurd meaning (it’s quite understandable). It’s just boring. It has a message, but the message isn’t interesting. As a story Rainy Day is a failure because the message conveyed isn’t worth the work required to extract it.

So what makes a message worth the effort? What makes a message interesting?

Certain topics are inherently interesting. Sex and danger—both of which appeal to primitive drives—arouse interest in almost any context. Novelty and new ideas are also interesting—mankind has retained curiosity as one legacy of his primate heritage. Useful information is also interesting. An article on how to reduce your tax bill is likely to interest you for this reason. Still other topics appeal only to some readers. Presumably the reader of this book is interested in artificial intelligence. Or perhaps you are a member of my family, and you interests are aroused for other reasons. There are many ways a story can be interesting.

Sometimes life provides an interesting message. The author of Adrift (Callahan, 1986) was lost at sea for 76 days without food or water. The story of his experience is interesting because he faced danger, invented novel solutions to his problems, and learned useful information about survival under the most difficult of circumstances. But when life doesn’t provide an interesting story, the author must create his own message:

A storyteller must fashion his story to convey an interesting message.

Finding, formulating, and conveying an interesting message is one of the reasons that storytelling is such a difficult task. To build a computer program that
can tell stories, we must build a model of communication. The computer program must (1) know what an interesting message is, (2) be able to select a message to convey, and (3) be able to create a story that illustrates the message. Building a model of “messages” and designing the processes that can illustrate a message is one of the challenges of creating a program that can tell stories.

1.2.3 Creativity

In literature, as in all the arts, there is a premium placed on creativity. To be art, a work must be new and different in significant ways. No publisher would accept a copy of *Romeo and Juliet* with only the names changed. Even an author who tells consistently interesting stories inevitably loses popularity if his stories are all very similar. One of the reasons that storytelling is so difficult is that the author is challenged to be creative. It isn’t enough to tell an interesting story; the author must also strive to make the story new and different.

A storyteller must be creative.

But being creative is hard. Even judging whether or not something is creative is difficult. Surely copying *Romeo and Juliet* with only the names changed is not creative. But what if an author copied *Romeo and Juliet* and changed the setting to, say, the west side of New York City? The musical *West Side Story* was a tremendous Broadway hit and award-winning movie. Was Leonard Bernstein being creative when he wrote *West Side Story*? Or does Shakespeare deserve the credit for that success?

Whether or not something is creative depends on the number and quality of its differences from similar works. But how can we distinguish inspiration from plagiarism? How can we judge when something has enough differences from previous work to be creative? And how can we determine if the differences are significant? These are just some of the problems in determining whether something is creative.

And if judging creativity seems difficult, being creative seems almost impossible. The creative person performs an almost miraculous feat: the bringing forth of something new and novel, something never before seen. The average person cannot write a creative story, paint a masterpiece, or invent a new device. The few people who are consistently and greatly inventive—William Shakespeare, Albert Einstein, Leonardo da Vinci, Thomas Edison—are revered as geniuses. And yet storytelling is an activity that demands creativity. No wonder then that it is so difficult to tell stories.

What is the source of creativity? The ancients believed that creativity was the work of a supernatural Muse who spoke into the artist’s ear. Although few artists today would profess to believe in a literal Muse, many do believe that creativity originates in processes that are beyond human comprehension. Is creativity a mystery that can never be explained by man, nor expressed in language? Or does creativity have a natural explanation in the cognitive processes of the human mind?

Many psychologists and cognitive scientists today believe that creativity is the result of cognitive processes that bring together pieces of old knowledge in new ways. But that is hardly a full explanation of creativity. To embody this model of creativity in a computer program requires addressing a myriad of difficult issues:

- How is knowledge organized and searched?
- How is knowledge combined to form new knowledge?
- How does creativity interact with other cognitive processes?
- Are there different types of creativity?
- Is imagination different from creativity?
- How does the creator recognize something new?
- How does the creator guide his creativity?
- How does the creator evaluate his creation?

The requirement to be creative is one reason that storytelling is such a difficult task for humans; the requirement to understand and model creativity is one of the reasons that building a computer program to tell stories is so difficult.

1.2.4 Art and Language

The sciences distinguish between content and presentation. We can and do speak of research as “important but poorly presented” or “polished but lacking substance.” In the arts, however, it is much more difficult to separate presentation from content. In the sciences, presentation is a secondary concern, needed only to communicate content. But in the arts, presentation is part of the content. Art demands good presentation in a way that science does not.

This is true in literature as in the other arts. An author is expected not only to present a meaningful, interesting, and creative story, but also a well-crafted story. At one level, this requires that the author use language well, following the rules of grammar and punctuation while accurately conveying his meaning. At another level, this requires that the author present his story beautifully, by using language in poetic ways and by using dramatic writing techniques such as characterization and foreshadowing. In addition to all the other goals he is trying to achieve, a storyteller must try to create an artistic story.

A storyteller must create stories that are aesthetically pleasing.

Of course, this further complicates the problem of creating a storytelling
program. To tell a story that meets literary standards as well as standards of comprehension, interest, and creativity, a storytelling program needs knowledge of both language and drama. To use language well, a computer storytelling program requires knowledge about words and what they mean, an understanding of grammar, and a model of how language is produced—how concepts are expressed in language. To produce a story that is artistically pleasing, a computer storytelling program requires knowledge about the structure and parts of a story, knowledge about dramatic writing techniques (including how they are applied and what they achieve) and a model of dramatic writing that captures how an author uses dramatic writing techniques to improve the artistic value of his stories.

1.3 MINSTREL: A Computer Model of Storytelling

Clearly, telling a story is very difficult. A good story must be understandable, interesting, creative, and artistic. Alone each of these goals is difficult to achieve; together they are truly formidable. It is no wonder that most adults are not good storytellers.

And as we have seen, the task of building a computer program to tell stories is even more daunting. Building a computer storyteller requires capturing all the knowledge a human author uses to tell a story, building models of storytelling, creativity, interestingness, and art, and executing the plans and processes an author uses to tell a story. If not outright impossible, this is at least an enormous job.

In this section, we give a brief overview of MINSTREL, a computer program that tells stories about King Arthur and his Knights of the Round Table. As one might expect from the difficulty of the storytelling task, MINSTREL is a large and complex program representing many years of research. It isn’t possible to give a detailed description of MINSTREL in just a few pages. Instead, we hope to give the reader a general overview of MINSTREL’s design and structure, and a preliminary glimpse into the kinds of knowledge and processes MINSTREL uses to tell stories. Hopefully this prelude will entice the reader deeper into this book to discover the inner workings of MINSTREL.

1.3.1 What is MINSTREL?

MINSTREL is a computer program written in Common Lisp that tells stories about King Arthur and the Knights of the Round Table. MINSTREL is about 17,000 lines of code, and is built upon a tools package called Rhapsody (Turner & Reeves, 1987) that is itself about 10,000 lines of code.

The stories MINSTREL tells are one-half to one page in length. MINSTREL begins with a small amount of knowledge about the King Arthur domain, as if it had read one or two short stories about King Arthur. Using this knowledge, MINSTREL is able to tell more than ten complete stories, and many more incomplete stories and story fragments. In addition to storytelling, MINSTREL has been used to invent mechanical devices and to solve planning problems.

1.3.2 A Story

When they first hear about MINSTREL, most people are curious to read a story that MINSTREL has written, so that they can judge for themselves whether or not MINSTREL is a competent author. Accordingly, we present here a representative story so that you can judge for yourself MINSTREL’s capabilities. This story and other stories throughout the volume are presented exactly as produced by MINSTREL, with the exception of the titles, which were added later by the author. Here then, is one of MINSTREL’s stories:

The Vengeful Princess

Once upon a time there was a Lady of the Court named Jennifer. Jennifer loved a knight named Grunfeld. Grunfeld loved Jennifer.

Jennifer wanted revenge on a lady of the court named Darlene because she had the berries which she picked in the woods and Jennifer wanted to have the berries. Jennifer wanted to scare Darlene. Jennifer wanted a dragon to move towards Darlene so that Darlene believed it would eat her. Jennifer wanted to appear to be a dragon so that a dragon would move towards Darlene. Jennifer drank a magic potion. Jennifer transformed into a dragon. A dragon moved towards Darlene. A dragon was near Darlene.

Grunfeld wanted to impress the king. Grunfeld wanted to move towards the woods so that he could fight a dragon. Grunfeld moved towards the woods. Grunfeld was near the woods. Grunfeld fought a dragon. The dragon died. The dragon was Jennifer. Jennifer wanted to live. Jennifer tried to drink a magic potion but failed. Grunfeld was filled with grief.

Jennifer was buried in the woods. Grunfeld became a hermit.

MORAL: Deception is a weapon difficult to aim.
The reader will probably have a number of immediate first impressions about this story:

- The story has a “point.”
- The story is understandable and consistent.
- The story is reasonably clever, particularly the deception and how it leads to grief.
- The use of English is not as polished as that of a human author.

Most people judge MINSTREL’s stories to be competent but not brilliant, equivalent to the kinds of stories people expect from children 10–15 years of age. However, the reader will soon realize that it is difficult to evaluate MINSTREL’s performance solely by looking at the stories it creates. As important as what MINSTREL creates is how it creates. Certainly The Vengeful Princess is less impressive if it was produced from canned text hidden in MINSTREL’s code. To understand MINSTREL we have to look deeper. We have to look at the processes and knowledge MINSTREL uses to create stories.

1.3.3 The Architecture of MINSTREL

MINSTREL is based on a model of the author as a problem solver. We are not accustomed to think of art as problem solving. The phrase problem solving brings to mind scientific disciplines, logic, and schoolwork. And yet there are few fundamental differences between creative domains such as storytelling, art and music and more mundane problem domains, such as science and day-to-day problem solving. In art as in day-to-day life, people have goals, find or create plans to achieve those goals, apply the plans, evaluate the results, and so on. To be sure, the arts are generally more creative than other problem domains. But anyone who has performed scientific research or jury-rigged a temporary fix to a household problem knows that creativity is both possible and necessary in science and day-to-day life.

In MINSTREL, storytelling is treated as problem solving. The same architecture and processes are used to write stories, to solve problems in the story world (such as how a knight can kill a dragon), and to invent mechanical devices. It is a basic tenet of this research that the problem solving process is invariant across problem domains, whether they be mundane or artistic, creative or noncreative:

The problem solving process is invariant across problem domains.

---

Figure 1.1 Case-Based Problem Solving

How then does MINSTREL solve problems? MINSTREL is a type of case-based reasoner (Reisbeck & Schank, 1989; Slade, 1991). The fundamental principle of case-based reasoning is that reasoners remember past problem solving situations (“cases”) and use that knowledge to solve current problems. One of the major values of the case-based reasoning paradigm is that it explains how reasoners can learn from experience: They remember what happens and reuse that knowledge when appropriate.

MINSTREL’s case-based problem solving process is illustrated in Figure 1.1. Problem solving proceeds in three steps: (1) The current problem is used to recall similar past problems; (2) The past solution is adapted to the current problem; and (3) The recalled solution is then applied to the current problem. Because there is nothing specific in this model to the storytelling problem domain, this same process can be used to solve both author-level storytelling problems and character-level planning problems, as well as problems in mechanical device invention.

MINSTREL’s model of problem solving is discussed in more detail in Chapter 3. However, one major shortcoming of this model should already be obvious. Case-based reasoners solve problems by reusing past solutions. A case-based storyteller would tell the same story over and over again—repeatedly solving the storytelling problem by reusing an old solution. Such a storyteller would be the opposite of creative. What do we need to add to a model of case-based reasoning to support creativity?
1.3.4 The Challenge of Creativity

The challenge of creativity is to find and use old knowledge in new ways to form a novel and useful solution to a problem. What processes and knowledge do we need to be able to create new problem solutions from old knowledge? How can we extend the case-based model of problem solving to include creativity? To begin answering these questions, let’s look at an example of creativity. The following story concerns a UCLA engineering professor:

One night, the professor and his wife were out late at a party. Returning home along a lonely road, the car they were driving slowly lost its electrical power. Looking under the hood, the professor discovered that the fan belt had broken.

After a perfunctory search in the trunk for a spare fan belt, the professor returned to the car and asked his wife to remove her pantyhose. The professor tied a series of knots in the pantyhose, and then tied the pantyhose into a loop somewhat smaller than the fan belt. He forced the loop over the crankcase pulley and the alternator pulley. The elasticity of the pantyhose kept it on the pulleys; the knots provide traction so that the improvised belt wouldn’t spin uselessly.

After giving the battery a few minutes recovery, the professor started the car and successfully made it home.

In this example, the professor brings together things he already knows—requirements for a fan belt, available materials, the properties of pantyhose, and knowledge of knots—to create a new and useful solution to the “fan-belt” problem. From everything the professor knew, he somehow picked out the knowledge that when combined in a new and unique way would result in a solution to his problem.

This process of finding and combining knowledge seems paradoxical. On the one hand, if the professor randomly grabs at old knowledge to try to create a new solution, he has an impossibly hard task in applying the knowledge he finds. Suppose the professor had recalled what he knew about grading papers, eradicating garden pests, and the rules of ping-pong and tried to apply that knowledge to the fan-belt problem. Surely he would fail. On the other hand, if the professor uses only knowledge that he knows how to apply to the current problem, he is unlikely to create a new solution. The professor’s knowledge of normal car repair procedures and the tools and parts used can be easily applied to the fan-belt problem. But it is unlikely to result in a new solution.

This is a classic “Catch-22” situation. The creator must somehow search everything he knows to find knowledge that can be adapted to create a new solution. But without knowing what knowledge can be adapted, how can the creator recognize what knowledge is useful? Conversely, once the knowledge has been found, the creator must somehow adapt it to the current problem. But without knowing how the knowledge is related to the current problem, how can the creator adapt it? The challenge of creativity is to define a creative process that (1) will find and recognize useful knowledge, and (2) know how to adapt that knowledge to create a new solution.

The MINSTREL solution is to integrate the search and adapt processes of creativity. By integrating the search and adapt processes, search can be guided by adaptation knowledge and adaptation guided by search knowledge. The search process finds only knowledge that it knows how to adapt; the adaptation process knows what adaptations to apply because it knows how the knowledge was found.

Creativity is an integrated process of search and adaptation.

In MINSTREL, the search and adaptation processes of creativity are integrated in heuristics called Transform-Recall-Adapt Methods, or TRAMs. Each TRAM bundles a search method with a corresponding adaptation. “Transform” takes a problem and changes it into a slightly different problem. “Recall” takes the new problem description and tries to recall similar past problems from memory. “Adapt” takes the recalled problem solutions and adapts them to the original problem.

One of MINSTREL’s TRAMs is TRAM:Cross-Domain-Solution. This TRAM is based on the idea that there are often enough similarities between two problem domains that solutions from one domain can be applied in the other. For example, military tactics can be applied to business problems—as evidenced by the popularity of A Book of Five Rings (Miyamoto, 1982) with both Japanese and American businessmen. TRAM:Cross-Domain-Solution suggests that a creative person can take advantage of this by translating a problem into a new domain (“Transform”), solving the problem in that domain (“Recall”), and then translating the solution back into the original domain (“Adapt”).

MINSTREL uses TRAM:Cross-Domain-Solution when telling a story called The Mistaken Knight. In the course of telling this story, MINSTREL has to create a scene in which a knight accidentally meets a princess. As it happens, MINSTREL doesn’t know any way in which a knight can accidentally meet a princess, so it must use creativity to solve this problem. To do this, MINSTREL uses TRAM:Cross-Domain-Solution.

The first step of the creative process is to search for new knowledge to apply to the current problem. TRAMs do this by transforming the current problem into a new problem, and finding the problem solutions that have been previously used for the transformed problem. These problem solutions are “new” in the
sense that they've never before been applied to the current problem. By transforming the current problem into a different problem, TRAMs search new areas of knowledge.

The "Transform" part of TRAM:Cross-Domain-Solution translates the original problem into a new problem domain. In this case, TRAM:Cross-Domain-Solution transforms the current problem ("A knight accidentally meets a princess") from the domain of medieval stories into the modern day-to-day domain ("A businessman accidentally meets a person"). TRAM:Cross-Domain-Solution does this by translating the original problem element by element, trying to find correspondences between the two domains. In this case, TRAM:Cross-Domain-Solution translates a knight into a businessman, leaves "accidental meeting" unchanged, and translates a princess into a person.

The second step of the creative process is to use the transformed problem as an index for recall, to see if the creator has ever encountered a similar problem before. In this case, the transformed problem recalls this story:

Walking The Dog

John was sitting at home one evening when his dog began scratching at the door and whining for a walk. John decided to take the dog for a walk. While they were out, John ran across his old friend Pete, whom he hadn't seen in many years. John realized that he would never have run into Pete if his dog hadn't wanted a walk.

The final step of the creative process is to adapt the recalled solution to the original problem. Because TRAM:Cross-Domain-Solution knows that the recalled solution was found by translating the original problem into a new problem domain, adaptation is easy. TRAM:Cross-Domain-Solution needs only translate the recalled story back into the original domain, creating this scene:

One day while out riding, Lancelot's horse went into the woods. Lancelot could not control the horse. The horse took him deeper into the woods. The horse stopped. Lancelot saw Andrea, a Lady of the Court, who was picking berries.

The resulting scene is new and original, illustrating how TRAM:Cross-Domain-Solution can create a new solution by adapting knowledge from another problem domain.

MINSTREL's TRAMs integrate the search and adapt processes of creativity. This ensures that (1) Any knowledge found will be useful, and (2) The creator will know how to apply the knowledge found. Thus MINSTREL's TRAMs provide a model of creativity which resolves creativity's seeming paradoxes.

MINSTREL's model of creativity is discussed further in Chapters 2 and 3. Those chapters address a variety of other issues in creativity, including:

- How TRAMs incorporated into the case-based model of problem solving.
- How multiple TRAMs can be used simultaneously to increase the power of creativity.
- How a problem solver can recognize a creative solution.
- How creativity can be integrated into a model of episodic memory to create an imaginative memory.
- Descriptions and discussions of the TRAMs MINSTREL uses.
- Descriptions of experiments with the MINSTREL model of creativity.

1.3.5 Author Goals in Storytelling

The previous section illustrated how MINSTREL solves problems by using case-based problem solver augmented with creativity heuristics. Let's return now to the particular problem of storytelling and look at the kinds of problem an author has to solve when he tells a story.

Earlier we identified four concerns of an author. An author wants to make his story (1) interesting, (2) understandable, (3) artistic, and (4) creative. We have already seen how MINSTREL achieves the last goal. How are the other three goals achieved?

MINSTREL has four classes of author-level goals corresponding to the major concerns of an author. They are:

- Thematic Goals
- Consistency Goals
- Drama Goals
- Presentation Goals

Thematic goals are concerned with the selection and development of a story theme. The story theme is the point or moral of the story. These goals assure that the story MINSTREL tells will have an interesting message. Consistency goals focus on creating a story that is plausible and believable. Drama goals are concerned with the use of dramatic writing techniques to improve the artistic quality of a story. These goals represent MINSTREL's desire to tell a story that is aesthetically pleasing. Finally, presentation goals are concerned with how a story is presented to the reader, and represent MINSTREL's desire to tell the story in a pleasing and effective way.

These four classes of goals combine to create a complete story. In the fol
16 CHAPTER 1

Following four sections we'll look briefly at each class of goals, what it contributes to a completed story, and how MINSTREL achieves each type of goal.

1.3.5.1 Theme in Storytelling

The theme of a story is the point, moral, or general truth that the story illustrates. Some stories, like fables, have an easily identifiable theme. Other stories, such as novels, may have several themes. In general, we assume that an author tells a story in order to present some point to the reader, or to illustrate some truth. We call this the theme of the story.

Themes help make stories interesting. Partly this is curiosity: If the reader realizes that a story has a purpose or point, he is curious to discover what it is. Partly it is self-interest. If the theme represents useful knowledge, such as a general truth about how the world works, the reader is interested in learning the theme so that he can use it to improve his life. Themes are thus one way an author can give his stories purpose and make them interesting.

MINSTREL's stories have very specific themes called Planning Advice Themes, or PATs. Each PAT represents a stereotypical planning situation and advice about how to handle the situation. Each PAT is a morsel of planning advice, and can often be summarized by an adage.

For example, the theme of The Vengeful Princess is summarized by the adage "Deception is a weapon difficult to aim." This theme is called PAT:Juliet, and is based on one of the themes in Romeo and Juliet. In the last act of Romeo and Juliet, Romeo takes a potion that makes her appear to be dead. Her intention is to deceive her family, but she deceives Romeo instead. Romeo, stricken with grief at the apparent loss of Juliet, kills himself. PAT:Juliet captures the mistake that Juliet made, and advice for planners considering a similar plan: "Be careful with deception plans because you may fool someone unintended."

MINSTREL's primary goal when storytelling is to tell a story that illustrates a particular Planning Advice Theme. By making this its primary storytelling goal, MINSTREL ensures that the stories it tells will (1) have a point or purpose, and (2) be interesting.

MINSTREL's plan for illustrating a theme involves creating story events that form a specific example of the planning advice the theme represents. This can be seen in The Vengeful Princess, where MINSTREL has created story events to illustrate the theme "Be careful with deception plans because you may fool someone unintended":

Jennifer wanted to appear to be a dragon [to fool Darlene]. Jennifer drank a magic potion. Jennifer transformed into a dragon...

Grunfeld was near the woods. Grunfeld fought a
dragon. The dragon died. The dragon was Jen-

These events illustrate the story theme by showing how Jennifer's deceit plan leads to grief when Galahad is fooled instead of Darlene. And by including an example of the theme in the story, MINSTREL makes the story purpose and interesting.

MINSTREL's use of themes is discussed in more detail in Chapter Among the issues addressed are:

- The structure and representation of Planning Advice Themes.
- How a theme is selected for storytelling.
- How the events that illustrate a theme are created.
- How new story themes can be invented.
- How story themes function as advice.
- Descriptions of the story themes that MINSTREL knows.

1.3.5.2 Consistency in Storytelling

As the reader may have noticed, the story events that illustrate the theme only a small part of the final story. The Vengeful Princess is about 30 sentences long, but only 10 of those sentences concern the story theme. Even though illustrating a theme is MINSTREL's primary purpose in storytelling, a good story requires that MINSTREL be concerned with many other goals. One of these goals is consistency.

A good story must be consistent at many levels. The author must have a deep understanding of both story form and the world in which the story takes place. The characters and the world should act consistently and predict MINSTREL tries to create stories in which the characters act as rational intelligent planners, in which the characters show the proper emotional reactions to events in their lives, and in which the story world is consistent and plausible. These concerns are apparent in several places in The Vengeful Princess.

To illustrate the story theme, MINSTREL creates a scene in which Gala kills a dragon (which turns out to be Jennifer). This story scene satisfies the moral requirement that Jennifer's deception leads to grief. But in other ways, the scene is inconsistent. There is no explanation of why Galahad killed the dragon, or how he reacts to the discovery that the dragon is really Jennifer.

To correct these problems, MINSTREL adds new story scenes. First MINSTREL adds scenes to explain why Galahad killed the dragon:

Grunfeld wanted to impress the king. Grunfeld wanted to move towards the woods so
that he could fight a dragon. Grunfeld moved towards the woods. Grunfeld was near the woods. Grunfeld fought a dragon...

MINSTREL uses knowledge about knights, what their typical goals are, and how they achieve those goals to create story scenes which explain why Galahad fights a dragon.

By detecting and correcting story inconsistencies such as this one, MINSTREL tells stories that are plausible and understandable. Story consistency is discussed in more detail in Chapter 6. Among the issues addressed are:

- How story consistencies arise.
- The types of planning inconsistencies MINSTREL can detect and correct.
- The types of story world inconsistencies MINSTREL can detect and correct.
- How emotions are modeled in MINSTREL.
- How emotional inconsistencies are detected and corrected.

1.3.5.3 Art and Drama in Storytelling

Another major concern of a storyteller is to create a story that has aesthetic appeal. By writing a story with literary values, an author helps make his story interesting and increases the emotional and intellectual impact of his story.

Human authors have many literary writing techniques: pacing, characterization, dialogue, suspense, foreshadowing, description, and many others. A look at the creative writing section of any bookstore will reveal that there are as many writing techniques as there are authors to expound them. Every human author develops a combination of literary writing goals and techniques that create a particular writing style.

It would be impossible to model all these techniques and their myriad combinations in MINSTREL. Instead, MINSTREL implements a few representative techniques: suspense, tragedy, characterization, and foreshadowing. This particular combination of techniques represents, if you will, MINSTREL's writing style.

One of the techniques that MINSTREL uses in creating *The Vengeful Princess* is tragedy. Tragedy is a literary form that evokes feelings of pity and regret in the reader, and often involves a character suffering a downfall because of a tragic flaw. MINSTREL has the ability to recognize when a story has the potential to be tragic, and has techniques it uses to emphasize the tragedy. The seed of a tragic situation occurs in *The Vengeful Princess* when Galahad accidently kills Jennifer. The tragedy here is that Jennifer’s temper leads her to seek revenge for a trivial reason, and eventually results in her own death.

MINSTREL has several plans for accentuating a tragic situation. One of these plans increases the impact of a tragic situation by making the tragedy occur at the hands of a loved one. This plan was used in the telling of *The Vengeful Princess* by making Galahad and Jennifer lovers:

Once upon a time there was a Lady of the Court named Jennifer. Jennifer loved a knight named Grunfeld. Grunfeld loved Jennifer...

...Grunfeld fought a dragon. The dragon died. The dragon was Jennifer...

The fact that Jennifer’s tragic flaw leads to her death at the hands of someone who loves her intensifies the tragic aspect of this story.

By applying dramatic writing techniques like tragedy, MINSTREL creates stories that have literary value and are aesthetically pleasing. MINSTREL’s dramatic writing goals are discussed in more detail in Chapter 5. The writing techniques described are:

- Suspense
- Tragedy
- Characterization
- Foreshadowing

1.3.5.4 Presenting the Story to the Reader

The final task of an author is to present his story to his readers. He must express the story in language, in a manner that is both pleasing and understandable. A good author will also fashion his story presentation to reflect and accentuate the purposes of his story.

To express a story in English, MINSTREL must accomplish several tasks. First, MINSTREL must select an order in which to relate the events of the story. In many cases, this ordering can be guided by the temporal ordering of story events. In other cases, such as foreshadowing, MINSTREL must make specific decisions about the order in which to present story scenes. Second, MINSTREL must generate the story events in English. This requires selecting words to express concepts, building up grammatical sentences, paragraphing, and many other tasks. Finally, MINSTREL may also create new story scenes in order to improve the presentation of the story.
In *The Vengeful Princess*, MINSTREL creates a story scene to introduce the reader to the main character of the story:

> once upon a time there was a Lady of the Court named Jennifer...

and story scenes to resolve character fates:

> Jennifer was buried in the woods. Grunfeld became a hermit.

These scenes are created to improve the story presentation. The introductory scene is created to ease the reader’s transition into the story by providing an immediate identification of the main character and the story genre. The final scenes are created to resolve the reader’s expectations about the fates of the story characters and create a sense of closure.

MINSTREL achieves a number of difficult tasks in presentation. These tasks include:

- How introduction scenes are created.
- How denouement scenes are created.
- How story events are ordered, both implicitly and explicitly.
- How paragraphing is used to reflect thematic structure.
- How natural language is generated from concepts.

Further information about how MINSTREL achieves its presentation tasks can be found in (Turner, 1993).

1.4 A Reader’s Guide

The remainder of this volume is divided into three sections.

Chapter 2 describes MINSTREL’s model of creativity, how creativity is used to augment case-based reasoning, and gives examples of MINSTREL’s creativity in planning and storytelling. Chapter 3 also discusses a variety of related issues, including errors in creativity and learning.

Chapter 3 introduces MINSTREL’s model of storytelling. Chapters 4, 5, and 6 describe MINSTREL’s author-level goals and plans for theme, drama, and consistency.

Chapter 7 presents an extended example of how MINSTREL tells a story. A detailed trace of MINSTREL creating a story is analyzed to reveal the exact sequence of author goals and plans that leads to a finished story.

Finally, Chapter 8 evaluates the MINSTREL model by comparing it to previous work in psychology and artificial intelligence and by analyzing MINSTREL’s performance as a storyteller and creator. Chapter 9 contains conclusions and some thoughts on future work.

2.1 Introduction

Creativity—the bringing forth of an original product of the human mind—is the pinnacle of human cognition. Artists, writers, and scientists are treated with intellectual awe, and the word itself brings to mind the roll of men whose discoveries changed their social and scientific cultures: William Shakespeare, Albert Einstein, Leonardo da Vinci, Thomas Edison.

And yet creativity has a mundane side as well. From small child to adult, we are all creative in every aspect of our lives. Faced with a problem we have never before encountered, we combine past knowledge in new ways to create a solution. We fix our cars using hangers and baling wire, invent jokes based on the latest domestic crisis, and make up bedtime stories for our sons and daughters. Far from being the sole province of extraordinary thinkers, the ability to create new solutions to problems is one of the cornerstones of human problem solving.

To understand human cognition, it is essential that we understand the processes of creativity: the goals that drive people to create and the mechanisms they use to create. And because creativity spans the intellectual spectrum from the highest peaks to everyday cognition, it is likely that studying creativity will provide special insights into human thought.

Understanding creativity also has practical value to computer scientists. Current computer programs have little adaptability. Faced with new situations, they act incorrectly or fail completely. Implementing the fundamental processes of creativity in computer programs has the potential to greatly increase the scope and power of the modern computer.
This chapter presents a process model of creativity. A case-based problem solver is augmented with evaluations that act as a creative drive, and a set of creativity heuristics called Transform-Recall-Adapt Methods (TRAMs). TRAMs create new solutions to problems by transforming problems into slightly different problems, solving the new problems, and then adapting any solutions found to the original problem. Repeated application of TRAMs enables the problem solver to make elaborate problem transformations by small steps, invent new solutions substantially different from old ones, and greatly extend the range of problems that can be solved.

The second part of this chapter discusses the implementation of this model in MINSTREL. Although MINSTREL is primarily a storytelling program, MINSTREL's model of creativity is independent of the storytelling domain. Consequently, the creative problem solving portion of MINSTREL can be applied to other domains and problem tasks. This chapter presents examples of MINSTREL's model of creativity applied to different problems, and discusses some of the important issues of modeling creativity.

2.2 Creativity and Problem Solving

The goal of this research is to develop a model of creativity and answer the question: What are the processes of creativity common to many problem solving domains? The first step to answering this question is to understand what it means to call the solution to a problem "creative."

According to psychologists, people recognize a solution as creative if it (1) has significant novelty and (2) is useful (Koestler, 1964; Wallas, 1926; Weisberg, 1986).

We all recognize that creative solutions must be original. They must be new and different from old solutions. But the differences must also be significant. If an artist were to paint the Mona Lisa in a red dress instead of a blue one, the resulting painting would not be considered creative, despite its differences from the original. Significant novelty distinguishes creative solutions from ones that are only adaptations of old solutions.

Usefulness is the second characteristic of a creative solution. We expect problem solvers to be capable: They must develop solutions that solve their problems. Replacing a flat tire with an air raft is novel but not creative, because it doesn’t effectively solve the original problem. Creativity must be purposeful and directed. Creative solutions must have bearing and utility on the problems to which they are applied.

Creative solutions are, therefore, both useful and significantly novel. How are these features reflected in the creative process? Consider the following anecdote concerning the 7-year-old niece of the author:

One day, while visiting her grandparents, Janelle was seated alone at the dining room table, drinking milk and eating cookies. Reaching for the cookies, she accidentally spilled her milk on the table. Since Janelle had been recently reprimanded for making a mess, she decided to clean up the spill herself.

Janelle went into the kitchen, but there were no towels or paper towels available. She stood for a moment in the center of the kitchen thinking, and then she went out the back door.

She returned a few minutes later carrying a kitten. The neighbor’s cat had given birth to a litter about a month ago, and Janelle had been over to play with the kittens the previous day. Janelle brought the kitten into the dining room, where he happily lapped up the spilled milk.

Most people find Janelle’s solution to her problem creative. The use of a kitten as an agent and the substitution of “consumption of milk” for “removal of milk” are significant differences from Janelle’s known solution to the “spilled milk problem”, and the success of the new solution shows its usefulness. Janelle’s example illustrates three important principles of creativity:

Creativity is driven by the failure of problem solving.

When Janelle could not find a towel, her old plans for cleaning up a spill became unusable, forcing her to invent a new solution. The need for creativity arises from the failure of problem solving (Weisberg, 1986).

Creativity is an extension of problem solving.

Janelle’s solution to the spilled milk problem arises as part of her problem solving process, and even the creative aspects of her solution—using a kitten as an agent and substituting consumption of the milk for removal of the milk—are broken down into subtasks in the manner of a classical problem-solving strategy (divide and conquer). Although some theorists (e.g., Koestler, 1964; Wallas, 1926) have suggested that creativity is a process fundamentally different from problem solving, there is ample evidence to indicate that creativity is an outgrowth or extension of problem solving (Weisberg, 1986).

New solutions are created by using old knowledge in new ways.

Creative solutions do not spring forth newborn from the head of Zeus; they make use of what the problem solver already knows. Janelle knew that the neighbors had kittens, that kittens like milk, and that the goal “consumption of milk” could subsume the goal of “removal of milk,” and she combined this information into a new solution by using her general knowledge about agents. The significant novelty of creative solutions arises from the problem solver’s
application of knowledge in a new way to the problem (Koestler, 1964; Weisberg, 1986).

The model of creativity presented in this chapter incorporates these principles into a model of case-based problem solving.

2.3 Case-Based Models of Problem Solving

MINSTREL's model of creativity is built on a case-based model of problem solving (Hammond, 1988; Kolodner, 1987; Schank, 1987).

Case-based reasoning systems are driven by an episodic memory of past cases rather than a base of inference rules or knowledge-intensive heuristics. Given a problem, a case-based problem solver (1) recalls a past problem with the same features and its associated solution, (2) adapts the past solution to the current problem, and then (3) assesses the result. To drive home from work, a case-based problem solver recalls a previous time he or she drove home and the route he used, modifies the route in light of changing road conditions (perhaps a particular street is closed for repair) and then assesses that route according to his or her general knowledge about driving.

This model is illustrated in Figure 2.1. Problem specifications are input at the left side, where they are used to recall similar past problems. The solutions from these past problems are then adapted to the current problem. Finally, the adapted solutions are assessed to determine if they are useful and meet other domain-specific considerations (e.g., a mechanical engineer might want to create efficient solutions). (A more complete review of case-based reasoning can be found in Slade, 1991.)

Case-based problem solving has several benefits. First, case-based problem solving is similar to the reasoning done by human experts (Riesbeck & Schank, 1989). Expertise depends on experience, and case-based reasoning captures this connection.

Second, case-based problem solving explains how a problem solver can use new experiences to extend his knowledge. Planning and storytelling systems that capture problem solving knowledge as static rules (i.e., Lebowitz, 1985; Meehan, 1976; Warren, 1978) require additional mechanisms to explain how that knowledge can be extended.

A final advantage of case-based problem solving is efficiency at solving routine problems. By recalling a past problem that shares all the features of the current problem, the problem solver is assured (1) that the solution from the recalled problem will apply to the current problem, (2) that adapting the recalled solution to the current problem will be simple, (3) that little or no assessment of the adapted solution will be necessary, and (4) that the recalled solution is very likely to work. Re-using past solutions under identical circumstances requires little effort and results in a high degree of success.

However, case-based problem solving has an obvious shortcoming: It fails when faced with a new problem. If a similar past problem cannot be recalled, the case-based problem solver has no way to discover, build, or create a new solution. Using old solutions over and over again is acceptable and efficient in many domains. It would be tiresome, for instance, to create a new route home from work each day. But when no old solution is known, a problem solver must be able to create a new solution. Furthermore, case-based problem solving has difficulty explaining domains such as art and literature. In these domains, problem solvers often reject solutions simply because they are old or repetitive.

To capture the creative process, case-based problem solving must be extended to include (1) a creative drive, and (2) processes that can combine old knowledge in new ways to create problem solutions.

2.4 Failure-Driven Creativity

People are conservative problem solvers. They expend the effort to create new solutions when familiar, known solutions fail (Weisberg, 1986). Janelle created a new solution to the spilled milk problem only because the lack of a towel prevented her from applying an old plan.

Old solutions may fail for a number of reasons. They may not solve the problem, or may solve the problem inefficiently. Or they may be rejected simply because they are old, as in literature and art, where a premium is placed on new ideas. Whatever the cause, it is the failure of old solutions that drives the problem solver to create a new solution. A model of creative problem solving needs a mechanism for detecting planning failures and using that to drive the creative process.

Case-based problem solving can fail at each of the three steps shown in Figure 2.1: (1) if a past problem situation similar to the current problem situation cannot be recalled, (2) if a recalled solution cannot be adapted to the current problem, or (3) if the adapted solution fails a domain assessment. In the
MINSTREL model of creativity, a failure at any of these steps causes MINSTREL to attempt to create a new solution to the problem. Like Janelle, MINSTREL is creative when it encounters a problem for which its past solutions fail.

Because MINSTREL is a model of storytelling as well as creativity, MINSTREL also implements an artistic drive to create. Unlike their counterparts in traditional problem solving domains such as engineering, artists sometimes reject solutions even when problem solving is successful. Artists create for the sake of creation; they find repetition of solutions boring and unacceptable. (In fact, all human problem solvers get bored with repetitive solutions, but particular emphasis is placed on this motivation in the arts.) MINSTREL implements the artistic drive to create as a boredom assessment.

The boredom assessment operates during the Assess step of problem solving (see Figure 2.1). The boredom assessment examines proposed problem solutions to determine if they’ve been used too many times previously, that is, have become boring. To do this, MINSTREL uses episodic memory.

Episodic memory is the autobiographical record of the events and experiences that make up a person’s personal history (Cohen, 1989; Tulving, 1972). MINSTREL uses a model of episodic memory based on work by Schank (1982) and Kolodner (1984), and elaborated and tested by Reiser, Black, and Abelson (Reiser, 1983, 1986; Reiser, Black, & Abelson, 1985), who term it the “context plus index” model. In this model, episodes are organized according to their distinctive differences. Two episodes with significant differences fall into different memory categories and will not be recalled together. Episodes with no significant differences fall into the same memory category and are recalled as a group.

To determine if a solution has become boring, MINSTREL indexes the solution in episodic memory and counts how many times similar solutions have been used. If the solution has been used more than a small number of times, it is judged boring and rejected.

To illustrate this process, suppose that MINSTREL is building a story scene in which a knight’s life is endangered. MINSTREL’s episodic memory contains scenes about King Arthur and his Knights. Some of these scenes are initially seeded into MINSTREL’s memory, as if MINSTREL had read several short stories about King Arthur. Others are from stories MINSTREL invented during earlier storytelling sessions. In this example, MINSTREL’s episodic memory contains only one scene. In this scene, a knight fights a dragon. To build a new scene in which a knight’s life is endangered, MINSTREL tries to recall a similar scene from a previous story. This recalls the scene in which a knight fights a dragon. The next time MINSTREL creates a scene in which a knight fights a dragon (perhaps again to solve the “build a scene in which a knight is endangered” problem), the boredom assessment will reject the scene. “Knights fighting dragons” has become boring, and MINSTREL will be driven to create a new way in which a knight can be endangered, even though problem solving succeeded in finding a useful solution to the original problem.

As MINSTREL tells stories and indexes them in memory, MINSTREL’s storytelling behavior changes. MINSTREL may tell stories about knights fighting dragons for a while, but these soon become boring and MINSTREL moves on to other topics. Thus, the boredom assessment models the artistic drive to create. MINSTREL, like any human faced with a repetitive task, becomes bored with the “known” solutions, and is driven to create new solutions. And because of its creativity heuristics, MINSTREL can invent new solutions when it becomes bored with old ones.

MINSTREL’s boredom heuristic is simple. It ignores the type of problem being solved and other constraints that might realistically affect an artist’s determination of whether a concept or solution should be reused. A storyteller, for example, will not want to create new solutions for every character action in a story. How a knight gets from place to place is probably unimportant, and any known solution will do, no matter how frequently it has been used in the past. And in other domains there will be similar considerations.

For storytelling, MINSTREL applies the boredom heuristic only when creating the story events that illustrate the plot of the story. The plot, which is the sequence of story events that illustrate the theme or most important point of the story, is the most important element of the story, and so MINSTREL strives to be creative when building the plot. Other elements of the story, such as events that were added to keep the story causally consistent, are of less importance, and so MINSTREL accepts noncreative solutions for these problems.

For any particular artistic domain, then, there will be other considerations in determining whether a concept is “boring.” But the basic consideration will remain the novelty of the concept: how often that concept or a similar one has been used in the past. The basic process behind MINSTREL’s boredom heuristic—using episodic memory to determine the novelty of an idea—supports this basic need and has the flexibility to be augmented into a more complete and realistic boredom assessment.

2.5 The Challenges of Creativity

By what processes are new solutions created? As Janelle’s example illustrated, people invent new solutions by (1) recalling knowledge not part of the known solutions to the problem, and (2) adapting that knowledge to create a solution (Weisberg, 1986). Four factors make this a difficult task.

First, the obvious source of knowledge about the current problem—similar
past problems—has been exhausted. Case-based problem solving tries to find solutions indexed under similar past problems. But the creative process begins when problem solving fails; the solutions indexed under the current problem have already been tried and rejected. To discover a new solution, the problem solver must find knowledge not indexed under the current problem. The difficulty is knowing where in the space of episodic memories to seek knowledge. Knowledge grabbed willy-nilly from memory will be unlikely to apply to the current problem.

New solutions require knowledge not indexed under the current problem.

Second, the problem solver may not even have any incorrect or partial solutions on which to base his problem solving efforts. If the current problem is sufficiently different from past problems, then case-based problem solving may have failed to recall any similar past problems and their associated solutions. Thus it is difficult for the creative problem solver to search for new knowledge in the solution space, because he may not have even an incorrect solution from which to start. However, he will always have his problem description, so he can search the problem space. In general, the problem solver must find new knowledge by searching the problem space. The problem solver must change the current problem into some new problem, in hopes of finding useful knowledge in the recalled solutions to the new problem. This casts the problem of creativity in a new light: The first step to creativity is changing the problem, not the solution.

Creativity involves recasting the problem.

Third, the complexity of adapting the discovered knowledge may overwhelm the creative effort. If Janelle had attempted to adapt her plan for the “getting to school” problem to the “spilled milk” problem, the difficulty of the adaptation task would be insurmountable. (How can knowledge about riding a school bus be applied to the problem of cleaning up spilled milk?) Trying to solve a difficult problem by substituting an impossible problem is a poor strategy. Somehow the creative process must limit the complexity of the adaptation process.

Adaptation must avoid too much complexity.

Finally, the creative process must be capable of discovering solutions substantially different from the original solution. There is a need to limit the creative process to simplify the search for new knowledge and the adaptation problem, but there is also a need to find the knowledge needed to created an original solution. Some mechanism must exist that will enable the creative problem solver to find necessary knowledge, even when it is conceptually distant from the original problem.

Creativity must be capable of “leaps” to new problem solutions.

The challenge of creativity research is to find a cognitive mechanism that not crippled by these requirements. The creative process must be able to (1) find useful knowledge by searching the problem space, (2) limit the adaptation task and (3) discover solutions substantially different from the original solution.

2.6 MINSTREL’s Creativity Heuristics

The MINSTREL model of creativity is based on creativity heuristics that associate problem transformations with specific solution adaptations. To search the problem space for possible new solutions, MINSTREL begins at the original problem and applies small transformations that create new, slightly different problems. If one of these new problems can be solved, the associated solution can be used to create a new solution to the original problem.

Suppose, for example, that MINSTREL is trying to invent a method for knight to kill a dragon. Figure 2.2 illustrates the search space for this problem. At the center is the original problem, and around it are similar problems. To discover a new solution to the original problem, MINSTREL applies a problem transformation, jumping from the original problems to one of the nearly problems. MINSTREL now tries to solve this new problem. If MINSTREL is suc
A MODEL OF CREATIVITY 31

this problem specification by mapping it into the modern domain. Elements of the original problem specification are mapped to corresponding elements in the new domain. "A knight" becomes "a businessman" and "a princess" becomes "a friend." The new problem specification is "a businessman accidently meets a friend" and recalls this story:

Walking The Dog

John was sitting at home one evening when his dog began acting strange. The dog was scratching at the door and whining for a walk. Finally, John decided to take the dog for a walk. While they were out, John ran across his old friend Pete, whom he hadn't seen in many years.

This story is adapted to the original problem by mapping the story back into the King Arthur domain, creating the scene in which Lancelot's horse leads him to Andrea. The resulting scene is creative—novel and useful—because TRAM:Cross-Domain enabled the problem solver to (1) discover knowledge previously unconnected to the original problem, and (2) apply that knowledge to create a new solution.

2.7 The MINSTREL Model of Creative Problem Solving

Figure 2.4 illustrates how TRAMs are integrated into the case-based problem solving model. The Recall and Adapt steps of the basic problem solving model are augmented with a pool of TRAMs. During problem solving, a TRAM is selected from this pool and applied to the original problem. If the TRAM succeeds in discovering a solution and adapting it to the original problem, then problem solving succeeds. If problem solving fails, then the current TRAM is discarded, another selected from the pool of available TRAMs, and the cycle repeats.

2.7.1 Transform

The first step of the augmented problem solving model is to transform the original problem. In this step, the Transform portion of one of MINSTREL's TRAMs is applied to the original problem description, changing it into a new problem. This step is a new addition to the model of case-based problem solving. It is this transformation that permits MINSTREL to search the problem space for new knowledge, a capability that normal problem solving lacks.

Of course, a problem solver needs to be creative only if problem solving fails. So before applying a transformation to the original problem specification, the problem solver should try to solve the original problem.
2.7.2 TRAM Selection

When TRAM:Standard-Problem-Solving fails to find a solution, a new TRAM must be selected from the pool of available TRAMs to begin MINSTREL’s effort to create a new solution.

The TRAMs in the pool of available TRAMs are organized according to the types of problems to which they apply. Some TRAMs are general, and can modify any problem description. Other TRAMs are specific to certain types of problem descriptions. To select a TRAM for the current problem, MINSTREL uses the problem specification to select the applicable TRAMs from the pool of available TRAMs.

MINSTREL indexes the pool of available TRAMs using episodic memory. Just as episodic memory contains past problems and their associated solutions, episodic memory also contains past problems and associated TRAMs. Finding the TRAMs that apply to a particular problem thus becomes a matter of recall. MINSTREL uses the current problem as an index to episodic memory, and finds similar past problems and their associated TRAMs. TRAMs that apply to many different types of problem are associated with very general problem descriptions; TRAMs that apply to specific types of problems are associated with more specific problem descriptions. By recalling all of the problem descriptions that match the current problem, MINSTREL gathers all the applicable TRAMs.

From the applicable TRAMs, one TRAM is selected randomly. The reason for using random choice is straightforward. Creativity heuristics search the problem space for useful knowledge to apply to the current problem. But prior to actually performing that search, the problem solver cannot know where the useful knowledge lies. In this sense, the applicable TRAMs are indistinguishable. Hence the problem solver has no reason to prefer one heuristic over another, making random selection a reasonable algorithm.

Another possible algorithm for selecting between competing TRAMs is to use past creativity experience to guide TRAM selection. For example, the problem solver might want to use TRAMs according to how frequently they’ve been successful in past problem solving situations. Or, since that might lead to repetition, the problem solver might want to try heuristics that haven’t been frequently used. Or the problem solver might want to try the most recently successful heuristic again, to reinforce the value of newly learned heuristics. All of these are interesting strategies.

Although MINSTREL does not currently use past creativity experience to guide TRAM selection, the use of episodic memory to organize TRAMs provides support for these types of strategies. As TRAMs are used and indexed into episodic memory, the problem solver retains knowledge about the usage of creativity. However, what form a cognitively valid record of creativity would take remains an open question.
2.7.3 Recall

The second step in the MINSTREL model of creative problem solving is recall of similar past problems. If similar past problems can be recalled, the associated solutions are passed on to the adaptation step. The index for recall is the problem specification. For this reason, episodic memory must be organized by past problems, and the recall process must be able to take a possibly incomplete problem specification and recall similar past problems.

MINSTREL’s model of episodic memory is based on the “context plus index” model (Kolodner, 1984; Reiser, 1986; Schank, 1982). Context plus index models of memory organize a specific episode according to its general context (i.e., a plane trip) and the distinguishing features of the episode (i.e., taking Pan-Am, flying to Central America). In the MINSTREL model, problem types correspond to different contexts, and the specific features of each problem are used as indexing features. This permits MINSTREL to organize memory according to past problems, and to recall matching problems when given a problem specification.

2.7.4 Adaptation

The third step of MINSTREL’s creative process is adaptation. Past solutions to problems are passed to the adaptation step, where the Adapt portion of the controlling TRAM modifies them for use on the current problem.

In general, the problem of adapting knowledge—even useful knowledge—to a new problem is very difficult. Consider what a problem solver must do to adapt the story “Walking the Dog” to the problem of “Lancelot meets Guinevere unexpectedly.”

Walking The Dog

John was sitting at home one evening when his dog began acting strange. The dog was scratching at the door and whining for a walk. Finally, John decided to take the dog for a walk. While they were out, John ran across his old friend Pete, whom he hadn’t seen in many years.

The problem solver must first fully understand this story, so that he can recognize that it is also an unexpected meeting. Then he must determine what its relationship is to the original problem, to know how to apply this knowledge to it. He must realize that it is in another problem domain, and determine what that domain is. Finally, he must build a mapping from this domain to the original problem domain and translate the story.

In MINSTREL, though, the problem of adaptation is greatly simplified by associating adaptations with specific problem transformations. Because each adaptation is applied only to problem solutions that arose from a particular problem transformation, there is no need to fully understand the problem solution and determine its relation to the original problem. The relation of the recalled solution to the original problem is fixed by the Transform portion of the TRAM. Instead, the adap portion of each TRAM needs only do the final work of adaptation—to make the necessary changes to the recalled solution.

For example, in TRAM:Cross-Domain-Solution the Adapt step need only apply the cross domain mapping generated during the Transform step in reverse upon the problem solution. This translates the recalled solution back into the original problem domain, making it suitable for the original problem. The Adapt step does have to understand the recalled solution, or determine its relationship to the original problem.

2.7.5 Assessment

The fourth step of MINSTREL’s creative process is assessment. The purpose of the assessment step is to evaluate a proposed solution in light of acceptance criteria specific to a particular domain. For example, a mechanical engineer might evaluate his designs in terms of their efficiency, what kinds of materials they use, and so on. Although the problem solving process tries to create solutions that fit the original problem specification, it may inadvertently fail (see the discussion of “Creativity Errors” in this chapter), or there may be additional criteria that cannot be easily expressed in the problem specification. In these cases, domain assessments can be used to catch faulty solutions.

A special assessment for artistic problem domains is the boredom assessment. The boredom assessment rejects solutions that are too similar to previous solutions, and models the drive for originality in the arts. The boredom assessment is an example of an assessment that embodies a criteria that is difficult to express in the original problem specification—namely, that the solution be original.

When a proposed solution fails a domain assessment, problem solving fails, the active TRAM and the proposed solution are discarded, and another TRAM is selected. The problem solving process then repeats under the control of the new TRAM.

One shortcoming of this algorithm is that it discards the proposed solution, which may be almost entirely correct. Re-inventing the correct parts of that solution may take a great deal of effort. An alternative possibility is to “repair” proposed solutions which fail a domain assessment.

For example, if a domain assessment for mechanical invention notices that a device has a redundant component, it can repair that design by removing the redundant component. Another possibility is to use problem solving recursively to repair a faulty solution. If a domain assessment in mechanical invention
notices that a device lacks a power source, it can repair this problem by recursively using problem solving to invent a power source.

MINSTREL is capable of both these types of repair. Domain assessments can directly manipulate the proposed solutions or call problem solving to create a correction. Whether a proposed solution should be rejected or repaired depends both upon the cost and efficacy of the repair technique.

First, repair should be undertaken only when it provides a cost savings over rejecting the faulty solution and finding a new solution. Repairing a solution with major faults may be much more expensive and time-consuming than simply throwing out the faulty solution and finding an entirely new one. For a mechanical device lacking a power source, inventing a power source may be a very difficult problem that would take more time and effort than simply finding a different solution to the original problem. If this is the case, it would be better to reject the proposed solution rather than repair it.

Efficacy of repair can also be a factor. Consider the earlier example in which a repair removes a redundant piece from a mechanical device. Is that always a correct and safe change to the design? Only insofar as the repair understands the device design. Perhaps the redundant pieces was added to the design by problem solving in order to balance the weight of the device. In general, a repair heuristic will not be as knowledgeable and powerful as problem solving, and so will sometimes make errors.

Deciding the likely costs of repair vs. rejection, or determining whether a repair will be efficient without introducing more serious faults are difficult issues. Currently MINSTREL's boredom assessment uses rejection, while the two assessments used in mechanical invention use repair, but these are simply design choices that are not founded on any deep understanding of the issues of repair vs. rejection. For further discussion of the role of repair in planning, see Hammond (1988).

2.8 Summary of Creativity Model

Earlier we identified the three challenges of creativity as: (1) finding useful knowledge by searching the problem space, (2) limiting the adaptation task, and (3) discovering solutions substantially different from the original solution. The MINSTREL model of creativity answers these challenges by associating problem transformations with corresponding solution adaptations. Problem transformations permit MINSTREL to search the problem space, and because each problem transformation has an associated, specific solution adaptation, the complexity of adaptation is eliminated. And this adaptation of knowledge from other problems leads to the creation of new solutions with substantial differences from previous solutions.

2.9 Leaps of Creativity

TRAMs find new solutions to problems by making small changes in the problem description and corresponding small adaptations to any discovered solutions. In light of the definition of creativity as a solution with a "substantial" difference from past solutions, this may seem counterintuitive. Why not use creativity heuristics that make large changes to the problem descriptions? There are several reasons, which are founded in our current understanding of human cognition.

First, MINSTREL is an integrated model of problem solving and creativity, in which creativity is an extension of problem solving. In most problem solving situations, there is no need for powerful creativity. Most problems are solved using standard solutions or past solutions that have been only slightly adapted. It is only in rare cases that a problem solver must invent a solution substantially different from a past solution.

Consequently we expect creativity heuristics to be very efficient at discovering small adaptations while still capable of larger adaptations. By using heuristics that make small adaptations, MINSTREL is efficient at the types of simple problem solving and creativity that make up the bulk of problem solving situations. (The question of whether MINSTREL is capable of larger adaptations is discussed below.) But there are other reasons to use small adaptations as a basis for creativity.

One advantage of creativity heuristics that make only small changes to a problem description is that they are more likely to find useful knowledge. Slightly different problems are good sources of useful knowledge because they share many of the same constraints as the original problem, and their solutions are likely to have some applicability to the original problem. Examining slightly different problems constrains the search task to an area of the problem space that is localized and fertile.

A second advantage is that small adaptations are easier and more likely to be successful than large adaptations. Adapting a very different solution to a new problem (i.e., adapting the "getting to school" solution to the "spilled milk" problem) requires a great deal of knowledge and effort and is likely to fail no matter what the expenditure. Adapting a solution that has only small differences (i.e., adapting the "spilled juice" solution to the "spilled milk" problem) requires less knowledge and effort and is more likely to succeed.

So there are several reasons to use small adaptations. But are small adaptations capable of discovering more creative solutions?

Sometimes even simple heuristics are capable of discovering unique solutions. Consider the heuristic to "substitute an agent" which Janelle used to create a solution involving a kitten. Although the heuristic itself is simple and commonplace, in this case the solution it discovers is surprisingly creative.

But even if creativity heuristics taken singly are not capable of major
discoveries, they can be when taken in combination. The strategy of small problem transformations tends to find solutions with small differences from the original problem. But if a single heuristic does not find a new solution, transformations can be repeatedly applied until substantially different solutions are discovered. And because the adaptation of the created solution is done in many small, simple steps, the process of adaptation remains simple.

Leaps in creativity result from combinations of small modifications.

How can several creativity heuristics be applied to a single problem? In MINSTREL, this is achieved through a mechanism called imaginative memory.

2.9.1 Imaginative Memory

The central step of MINSTREL's creative problem solving model is recalling a solution from episodic memory. But recall itself can be viewed as a kind of "problem solving." What happens if creative problem solving is used to solve the "recall problem"?

To do this requires replacing the Recall step of creative problem solving with a recursive call to creative problem solving with the problem specification “Find something in episodic memory that matches these features.” Of course, this could lead to endless recursion: at each level of problem solving, the Recall step calls problem solving again. To terminate this endless recursion, TRAM:Standard-Problem-Solving is modified so that it will continue to use episodic memory for recall. Since TRAM:Standard-Problem-Solving is always the first TRAM used by creative problem solving, this means that the first attempt at recall at each level of recursion will use episodic memory instead of recursing to another level of problem solving. The result is that creative problem solving first tries to recall a solution from episodic memory. If that fails, it recursively calls creative problem solving to solve the "recall" problem. This process is illustrated in Figure 2.6.

Now when a problem solver needs to recall something, TRAM:Standard-Problem-Solving is the first TRAM used, and passes the recall features unchanged to episodic memory. If an episode that matches the recall features is found, problem solving succeeds. Because TRAM:Standard-Problem-Solving is always the first TRAM used and continues to use episodic memory normally, recall behaves as expected when an episode exists that matches the recall features.

Something more interesting happens when the Recall step of TRAM:Standard-Problem-Solving fails. If TRAM:Standard-Problem-Solving cannot find an episode in memory that matches the recall features, problem solving fails and a new TRAM is selected. This TRAM modifies the recall features and recursively calls the problem solving process with the new recall features.

The first TRAM used on the recursive call is TRAM:Standard-Problem-Solving. If the new features recall an episode, the episode is returned to the previous recursion of problem solving, where it is adapted to the original problem by the previous TRAM, and recall succeeds. But because the recalled episode was changed by the Adapt portion of the previous TRAM, it is no longer the episode that was found in memory.

Recall has succeeded in a strange way: by recalling an episode that does not exist in episodic memory. Episodic memory has become imaginative. When an appropriate episode exists, it is recalled. When no appropriate episode exists, recall uses creativity heuristics to "imagine" an appropriate episode.

Treating recall as problem solving also enables the problem solver to apply multiple TRAMs to a problem. Each time recall fails the recursive use of creative problem solving will apply another TRAM to the recall features. In this
way, a number of TRAMs can be successively applied to a problem. Each TRAM changes the problem in only a small way, but the cumulative effect may be large, enabling the creative problem solver to discover new solutions significantly different from known solutions.

There are two advantages to imaginative memory. First, it provides a simple and powerful mechanism for the repeated applications of creativity heuristics to a problem. Each time the Recall step of problem solving fails, imaginative memory will recurse and apply a new creativity heuristic. If no useful knowledge can be found in the problem space near the original problem, repeated problem transformations will move the creative problem solver into more distant areas of the problem space.

More importantly, imaginative memory implements creativity at a low cognitive level. By embedding creative problem solving in the recall process, imaginative memory makes creativity transparently available to any cognitive mechanism that uses episodic memory for reasoning. By integrating creativity into the foundation of the cognitive process, imaginative memory increases the reasoning power of all cognitive mechanisms.

2.10 Integrated Model

Figure 2.7 illustrates how the boredom assessment, Transform-Recall-Adapt Methods and imaginative memory are integrated with case-based reasoning to form a complete model of creative problem solving. The three steps of case-based reasoning (Recall, Adapt, Assess) have been augmented with a Transform step. An active TRAM controls the Transform and Adapt steps. Initially this is TRAM:Standard-Problem Solving, which is simply the strategy of recalling a similar past problem and using the solution from that problem. The Assess step applies a pool of assessments to proposed solutions. In creative domains, this includes the boredom assessment, which rejects solutions that are too similar to past solutions. The Recall step uses imaginative memory (a recursive call to problem solving) except when controlled by TRAM:Standard-Problem Solving.

The problem solving cycle begins when a problem description enters the recall step. Initially TRAM:Standard-Problem Solving is in control. The original problem description is used to recall similar problem solving situations from episodic memory. If recall succeeds, the recalled situations are passed to the Adapt step. Under TRAM:Standard-Problem Solving, no adaptation is needed because the recalled solutions are very similar to the original problem, so the recalled solutions are passed along to the Assess step. In the Assess step, all active assessments are applied to the recalled solutions, and if a solution passes all the assessments, it is output as a solution to the original problem. This is the normal problem solving cycle.

If TRAM:Standard-Problem Solving fails, either because no solutions were recalled or because the recalled solutions failed some assessment, TRAM:Standard-Problem Solving is discarded and a new TRAM selected. The selection of a TRAM is based on the type of problem being solved and the TRAMs previously used.

The selected TRAM transforms the original problem specification, creating a new problem specification. The new problem specification is passed to imaginative memory. If recall succeeds, the recalled solutions are passed to the adapt step, where the active TRAM applies a specific adaptation which reverses the problem transformation. If recall fails, then imaginative memory recursively applies a second TRAM, and creative problem solving repeats.

Figure 2.6 illustrates the recursive use of creativity problem solving. When recall from episodic memory fails, imaginative memory resolves to a recursive use of problem solving. This continues until recall from episodic memory succeeds (or a processing limit is reached). The solution from each level is passed back to the previous level, where it is adapted, assessed, and passed up again. Eventually the solution reaches the top level, at which point it has been adapted to the original problem.

At each step, adapted solutions are assessed by domain assessments and, if appropriate, the boredom assessment. If a solution passes all assessments, problem solving succeeds. If all solutions fail, the active TRAM is discarded, a new TRAM selected, and the Recall-Adapt-Assess cycle repeats.
2.11 Examples of Creativity

The model of creative problem solving presented in this chapter has been implemented in a computer program called MINSTREL. The next two sections show how MINSTREL uses creative problem solving to discover new solutions for a simple planning problem in the domain of King Arthur, and to create a scene for a story. These examples should give the reader a better understanding of the TRAM model of creativity.

2.11.1 Suicide Example

In this example, MINSTREL is trying to discover a way for a knight to commit suicide. Initially, MINSTREL knows nothing about suicide, but does know about killing dragons and drinking a potion to become ill. Using this knowledge and creative problem solving, MINSTREL discovers three methods of suicide and invents the notion of “poison.”

2.11.1.1 Representation

MINSTREL uses a schema-based representation language called RHAPSODY (Turner, 1985). Goals, actions, and states of the world are represented as schemas; instances of these schemas make up the episodes in MINSTREL's memory and the elements of the stories MINSTREL tells. Each schema has named slots which contain schema information. Goal schemas, for example, have slots for the type of the goal and the actor of the goal. Schemas can also have named links to other schemas. Goal schemas typically have links to plans, and to the states that achieve the goals. Schema names begin with an ampersand (&) and schema instances are given either descriptive names (such as &KNIGHT-FIGHT) or generated names based on the schema type (such as &GOAL.112). For a more complete discussion of MINSTREL's representation, see (Turner, 1985).

2.11.1.2 The Problem

Figure 2.8 illustrates MINSTREL's representation of the suicide problem. &HUMAN.12 is an instance of the human schema which represents the knight. The type slot of a human instance indicates the character's major role in the King Arthur world, and illustrates how MINSTREL uses schema slots to instantiate particular schema instances. The knight has a goal (&GOAL.11) which will be achieved by the knight being dead (&STATE.8). The plan for this goal (&ACT.4) is currently uninstantiated. MINSTREL's goal in this example is to instantiate &ACT.4 as an action or series of actions that will achieve the knight's goal of committing suicide.

2.11.1.3 Initial State of Episodic Memory

All of MINSTREL's knowledge of the King Arthur domain is contained in episodic memory. MINSTREL's creativity heuristics have general knowledge about goals, plans, and states of the world, but specific knowledge about the goals, plans, and actions of characters in the King Arthur domain is deduced from the contents of episodic memory.

At the beginning of this example, MINSTREL knows nothing about how a knight might kill himself. Initially, MINSTREL's episodic memory contains only these two episodes:

**Knight Fight**

A knight fights a troll with his sword, killing the troll and being injured in the process.

**The Princess and the Potion**

A lady of the court drank a potion to make herself ill.

---

4 I use the term *invent* to indicate a concept that is new to MINSTREL, if not necessarily new to the reader.
Figure 2.9 shows the schema representation of "The Princess and the Potion." &ACT.14 represents the action of Lady Andrea quaffing a potion. &STATE.17 represents the intentional outcome of that action—Lady Andrea becoming ill.

2.11.1.4 Trace

Figure 2.10 shows a trace of MINSTREL inventing three different methods of suicide. In this example, MINSTREL has been configured to exhaustively invent solutions to the suicide problem and present them in English as created. Normally MINSTREL generates copious debugging and tracing output. To spare the reader, the trace shown in Figure 2.10 has been edited to improve readability. Uninteresting portions of the trace and reasoning dead-ends have been deleted. These deletions have been marked in the trace with "[...]". Except for this editing, the trace appears exactly as generated by MINSTREL. The level of indentation of the trace reflects the level of recursive problem solving.

MINSTREL begins the example shown in Figure 2.10 by generating the initial problem specification in English: "A knight named John did something. John died." This is an English description of the schema representation shown in Figure 2.7. MINSTREL's English descriptions of schemas are produced by a phrasal generator (Reeves, 1989; Zernik, 1987).

The problem specification is followed by a trace of the problem solving cycle. As each new TRAM is applied, the name of the TRAM (i.e., TRAM:GENERALIZE-CONSTRAINT) is printed out. When TRAM:Standard-Problem Solving is used to attempt recall from episodic memory, a message is printed out showing the recall index and what was recalled. The very first part of the trace shows TRAM:EXAGGERATE-SCALED-VALUE being applied to this problem and failing when nothing (i.e., NIL) is recalled from episodic memory.

![Diagram](image)

Figure 2.9 Representation of "The Princess and the Potion"
When a solution is discovered, MINSTREL prints a message to that effect and generates an English language description of the solution.

2.11.1.5 TRAM:Generalize-Constraint

The first TRAM that succeeds in discovering a solution to the suicide problem is TRAM:Generalize-Constraint. This TRAM suggests that a new solution to a problem can be found by removing a solution constraint, solving the new problem, and then adding the constraint back to the new solution. Figure 2.11 shows an informal outline of this heuristic.

In the suicide problem, the constraints available for generalization are the schema slot fillers of the problem specification (Figure 2.7): (1) the actor is a knight, (2) the object of the state is the actor, (3) the type of the state is health, and (4) the value of the state is dead. TRAM:Generalize-Constraint suggests recalling scenes in which one of these constraints has been generalized.

In this example, MINSTREL generalizes constraint (2). The original problem specification is “a knight kills himself.” TRAM:Generalize-Constraint generalizes this specification by removing the constraint that the knight kill himself and replacing it with the more general constraint that the knight kill something. This is indicated by the message “Generalizing :OBJECT on &STATE.112.” printed in the trace. The new problem specification is “a knight kills something.” This generalization recalls the “Knight Fight” episode:

1. Executing TRAM:GENERALIZE-CONSTRAINT.
   Generalizing :OBJECT on &STATE.112.
   Recalling &ACT.118: &KNIGHT-FIGHT.
   Adapting by replacing &MONSTER.15 with &HUMAN.12.
   TRAM succeeds: (&ACT.405).

In “Knight Fight,” a knight kills a troll by hitting it with his sword. This episode is adapted to the original suicide problem by reversing the original transformation. The troll corresponds to the generalized constraint that a knight kills “something.” To adapt this solution to the original problem, this more general constraint must be replaced with the original constraint—that the knight kill himself. Therefore the Adapt portion of TRAM:Generalize-Constraint replaces the troll with the knight, creating a scene in which a knight kills himself by hitting himself with his sword:

Minstrel invented this solution:
(A KNIGHT NAMED JOHN Fought HIMSELF BY MOVING HIS SWORD TO HIMSELF IN ORDER TO KILL HIMSELF *PERIOD* JOHN DIED *PERIOD*)

TRAM:Generalize-Constraint has used previous knowledge about how knights kill monsters to create a scene in which a knight kills himself. This process is shown graphically in Figure 2.12.

Three issues that must be addressed in TRAM:Generalize-Constraint are (1) how features are selected for generalization, (2) how the selected feature is generalized, and (3) how the selected feature is added back into the created scene.

To maximize the success of the recall step of TRAM:Generalize-Constraint, the feature chosen for generalization should be likely to lead to the recall of a scene. To achieve this, MINSTREL makes a broad generalization of each feature in the representation and attempts recall using the generalized episode. Every generalization that results in recall is added to a pool, and the problem constraint to be generalized is selected randomly from this pool.

MINSTREL uses two methods to generalize a feature. First, the feature can be completely removed from the problem specification. This is the broadest possible generalization, and is used when selecting the candidate pool. And while this method provides a good, quick indication of whether generalizing a particular feature is useful, it is so broad that it often leads to the recall of scenes which are difficult to adapt to the original problem.

For example, suppose that MINSTREL is creating a scene in which “a knight gives a princess something that makes her happy” and chooses to generalize the “princess” feature by removing it altogether from the problem specification. The new problem specification—“A knight gives somebody something that makes him happy”—can recall any episode in memory in which a knight

Figure 2.11 Informal Outline of TRAM:Generalize-Constraint
feature and the instantiations of the generalization. One such generalization is based on class hierarchies.

Classes group concepts that have many similarities. The “People” class groups a variety of characters—princesses, knights, kings, and hermits—that have many similar features. There can be many class hierarchies, and objects can belong to several classes. Knights, for example, are members of both the “People” class and the “Violent Characters” class. By generalizing problem features within classes, MINSTREL is more likely to find a useful reminding.

For this reason, MINSTREL’s implementation of TRAM:Generalize-Constraint uses class generalizations. In the suicide example, the “knight” feature is generalized to “a Violent Character.” This recalls the “Knight Fight” episode, in which a knight fights a troll in order to kill the troll, because trolls are also members of the “Violent Character” class. If this generalization had failed, MINSTREL would have generalized to the superclass (“Actors”), and if that generalization failed, TRAM:Generalize-Constraint would have failed. A portion of MINSTREL’s class hierarchy is shown in Figure 2.13.

The final step in TRAM:Generalize-Constraint is to adapt the recalled episode to the original specification. This is achieved by replacing the generalized feature value with the original feature value throughout the recalled episode. In the suicide example, the troll in “Knight Fight” is replaced by the knight throughout the recalled episode, resulting in a scene in which a knight kills himself by fighting himself with his sword.

Figure 2.14 illustrates TRAM:Generalize-Constraint as implemented in the current version of MINSTREL.

2.11.1.6 TRAM:Similar-Outcomes

MINSTREL discovers a second method for committing suicide by using both TRAM:Generalize-Constraint and a new heuristic, TRAM:Similar-Outcomes-Partial-Change. TRAM:Similar-Outcomes-Partial-Change suggests that if an action results in a particular outcome, it might also result in other, similar outcomes. For example, if MINSTREL doesn’t know anything about riding horses except that a knight once rode one to a castle, MINSTREL can use TRAM:Similar-Outcomes to guess that a knight might also ride a horse to some other destination.

In the suicide example, TRAM:Similar-Outcomes-Partial-Change recognizes that being killed is similar to being injured, and transforms the problem description from “a knight purposely kills himself” to “a knight purposely injures himself.” If MINSTREL can recall an action in which a knight purposely injures himself, it will be adapted to the current problem by replacing the injury with death. In essence, TRAM:Similar-Outcomes-Partial-Change “guesses” that an
action that is known to result in injury might also result in death. MINSTREL then tries to recall a scene in which “a knight purposely injures himself”:

[*...*]

Executing TRAM:SIMILAR-OUTCOMES-PARTIAL-CHANGE.
Transforming &DEATH to &WOUND.
Recalling &ACT.136: NIL.

However, this does not recall either of the episodes in MINSTREL's episodic memory (as indicated by “Recalling &ACT.136: NIL.”). "Knight Fight" is not recalled because the knight does not intentionally injure himself;

TRAM: Generalize-Constraint

Transform Strategy

1. For each feature in the scene specification, eliminate the feature and attempt recall from episodic memory. If recall is successful, then add the feature to a pool of acceptable generalizations.

2. Randomly choose a feature from the pool of acceptable generalizations ($feature).

3. Create a generalization of $feature based on the class membership. Attempt to recall or create a scene based on that generalization. If successful, pass the recalled scene ($recall) to the adapt step.

4. Otherwise, create a generalization based on the superclass, and attempt recall.

5. Otherwise, create a generalization by eliminating the feature, and repeat, and attempt recall.

6. Otherwise, fail this feature and attempt another feature from the pool of possible generalizations.

Adapt Strategy

1. Build a correspondence between the original scene specification ($original) and $recall by matching all the features in $original with the same features in $recall.

2. Add to this correspondence a mapping from the selected feature ($feature) to the instantiation of that feature in $recall.

3. Copy features that are present in $recall but missing in $original from $recall to $original, translating through the correspondence.

"Princess and the Potion" is not recalled because the actor is not a knight. Since recall fails, imaginative memory recurses and applies a new TRAM:
Executing TRAM:SIMILAR-OUTCOMES-PARTIAL-CHANGE.
Transforming &DEATH to &WOUNDED.
Recalling &ACT.136: NIL.

[TRAM Recursion: &ACT.136]
Executing TRAM:GENERALIZE-CONSTRAINT.
Generalizing :ACTOR on &ACT.138.
Recalling: &PRINCESS-POTION.
...TRAM succeeds: (&ACT.447).
...TRAM succeeds: (&ACT.447).

Minstrel invented this solution:
(A KNIGHT NAMED JOHN DRANK A POTION IN ORDER TO KILL HIMSELF *PERIOD* JOHN DIED *PERIOD*)

At this new level, MINSTREL applies TRAM:Generalize-Constraint to the description “a knight purposely injures himself” and generalizes the “knight” feature. “Knight” is generalized to “anyone,” which results in the new problem specification “Someone does something to purposely injure himself.” Note that this problem specification has been modified twice from the original specification, once by TRAM:Similar-Outcomes-Partial-Change, and once by TRAM:Generalize-Constraint.

The new problem description recalls “The Princess and the Potion” in which a lady of the court drinks a potion to make herself ill. The Adapt portion of TRAM:Generalize-Constraint then adapts this scene by replacing “lady of the court” (the generalized constraint) with “a knight” (the original constraint). This results in a scene in which a knight makes himself ill by drinking a potion.

The adapted scene is returned to the previous problem solving level, where TRAM:Similar-Outcomes-Partial-Change also adapts the scene, by replacing the illness with death. The adaptation reverses the transformation that turned “death” into “illness.” This results in a scene in which a knight kills himself by drinking a potion, fulfilling the original description “a knight kills himself.” Note that in the course of inventing this scene, MINSTREL has also invented the idea of poison—a potion that kills (vs. just making one ill).

(In fact, MINSTREL has invented the more narrow notion of “a potion which will kill a knight.” To apply this to other animate beings, MINSTREL will have to use creativity again to generalize about the actor of this action. MINSTREL does this by applying TRAM:Generalize-Constraint again.)

The main issue in TRAM:Similar-Outcomes is determining when two outcomes are interchangeable. MINSTREL has two methods for deciding this question. These are implemented as separate TRAMs called TRAM:Similar- Outcomes-Partial-Change and TRAM:Similar- Outcomes-Implicit.

TRAM:Similar-Outcomes-Partial-Change reasons that if an action can result in a partial relative change of a state then the action can also result in a complete change of the state. TRAM:Similar-Outcomes-Partial-Change is shown in

Figure 2.15. In this example, TRAM:Similar-Outcomes-Partial-Change reasons that something that makes someone ill (a partial negative change in health) might also kill them (a complete negative change in health).

Like many of MINSTREL’s creativity heuristics, TRAM:Similar-Outcomes-Partial-Change is a “common-sense” rule that captures reasoning that is often useful but not always correct. For example, extending a partial state change could be used to reason that because a man can lift a book in one hand he would also be able to lift an automobile in one hand. There are two things to be said about this type of error.

First, this type of error points out the value of simple, constrained TRAMs that make only small changes in problem descriptions. By making only small extensions to a problem solver’s knowledge, a TRAM is less likely to make an error in its extrapolation. In fact, MINSTREL’s version of TRAM:Similar-Outcomes-Partial-Change only extends state changes one additional “step” in a known direction. If state of health is represented by the scale “Excellent Good Normal Ill Dead,” and MINSTREL knows an action that changed a man’s health from Normal to Ill, then TRAM:Similar-Outcomes-Partial-Change can only extrapolate that to an action that changes a man’s health from Normal to Dead. TRAM:Similar-Outcomes-Partial-Change cannot extrapolate to an action that would take a man’s health from I11 to Excellent, or even from Excellent to Dead. Similar limits can be applied to states which do not have discrete representations, although MINSTREL does not currently handle this. By using simple, constrained creativity heuristics, MINSTREL reduces the number of reasoning errors it makes.

Incremental imaginative steps reduces errors in reasoning.

Second, even with restricted TRAMs, MINSTREL can still make reasoning errors of this sort. But this is to be expected: A creative problem solver should make errors. MINSTREL uses creativity to actively extend its knowledge. By

TRAM:Similar-Outcomes-Partial-Change

Transform Strategy

If the problem specification has an act that results in a partial relative change of a state in some direction, create a new specification in which the relative change is extended in the same direction.

Adapt Strategy

Replace the change of state in the recalled episode with a relative change of state copied from the original problem specification.

Figure 2.15 TRAM:Similar-Outcomes-Partial-Change
making good use of what it already knows, MINSTREL can often make accurate
guesses about what it doesn’t know. But sometimes it will err. The challenge
faced by a creative problem solver is to find creativity heuristics that are produc-
tive without an inordinate number of reasoning errors. The issue of creativity
errors is discussed in more detail in Chapter 8.

Imaginative reasoning will sometimes produce errors.

The second TRAM MINSTREL uses to determine when two outcomes are
interchangeable is TRAM:Similar-Outcomes-Implicit. TRAM:Similar-
Outcomes-Implicit reasons that two outcomes are interchangeable in every situa-
tion if it can recall any situation in which they are interchangeable. For exam-
ple, if MINSTREL can recall a scene in which a knight fought and killed a troll,
and another scene in which a knight fought and killed a dragon, MINSTREL can
use this knowledge to guess that trolls and dragons are generally interchange-
able. TRAM:Generalize-Constraint used a class hierarchy—an explicit represen-
tation of object similarities—to substitute one feature for another. TRAM:
Similar-Outcomes uses an implicit representation of similarities to substi-
tute one outcome for another.

Like TRAM:Similar-Outcomes-Partial-Change, TRAM:Similar-Outcomes-
Implicit is a heuristic that can sometimes err. Again, this is a direct con-
sequence of its function as an extrapolator of knowledge, and is to be expected in a
creative problem solver.

TRAM:Similar-Outcomes-Implicit

Transform Strategy

1. Create a new problem specification: An uninstantiated act that results
   in the state from the original problem specification. Use this to recall
different acts which can cause the result from the original problem.

2. Use episodic memory to recall other possible results of the actions
   collected in (1). Build a pool of these alternate results.

3. Create a new problem specification in which the act from the original
   problem specification results in a randomly-selected alternate result. Use
   this new specification for recall.

Adapt Strategy

1. Replace the alternate result of the recalled episode with the result
   copied from the original problem specification.

Figure 2.16 TRAM:Similar-Outcomes-Implicit

2.11.1.7 TRAM:Intention-Switch

MINSTREL’s final plan for suicide is discovered using TRAM:Intention-Switch.
This heuristic suggests that if the effect of an action was intentional it might just
as well have been unintentional. TRAM:Intention-Switch is illustrated in Figure
2.17.

In the suicide example, TRAM:Intention-Switch transforms the original
specification from “a knight purposely kills himself” to a “knight accidently
kills himself.” Recall on this new specification (&ACT.174) fails, because
MINSTREL’s episodic memory does not contain any episodes in which a knight
accidently kills himself:

[...]
Executing TRAM:INTENTION-SWITCH.
Recalling &ACT.174: NIL.
[TRAM Recursion: &ACT.174.]
Executing TRAM:SIMILAR-OUTCOMES-PARTIAL-CHANGE.
Recalling &ACT.178: &KNIGHT-FIGHT.
...TRAM succeeds: (&ACT.588).
...TRAM succeeds: (&ACT.588).

Minstrel invented this solution:
(A KNIGHT NAMED JOHN FOUGHT A DRAGON BY MOVING
HIS SWORD TO IT IN ORDER TO KILL HIMSELF
*PERIOD* JOHN DIED *PERIOD*)

Problem solving is used recursively, and TRAM:Similar-Outcomes-Partial-
Change modifies the current problem description “a knight accidently kills him-
self” by changing “kills himself” into something similar: “injures himself.”
The new problem description is “a knight “accidently injures himself.” This
recalls “Knight Fight”, in which a knight is injured while killing a troll.

TRAM:Intention-Switch

Transform Strategy

If an action in the problem specification intends a result, create a new
problem specification in which the same action unintentionally achieves
the result.

Adapt Strategy

Replace the unintentional result of the recalled episode with a similar
intentional result.

Figure 2.17 TRAM:Intention-Switch
Both TRAM:Similar-Outcomes-Partial-Change and TRAM:Intention-Switch adapt this recalled scene. TRAM:Similar-Outcomes-Partial-Change replaces “injures himself” with “kills himself,” and TRAM:Intention-Switch replaces “accidentally” with “purposely,” resulting in a scene in which a knight commits suicide by intentionally losing a fight with a troll.

Three things are interesting about this particular invention. First, although this particular TRAM is very simple, it results in a very novel and interesting plan—a knight purposely losing a fight to a dangerous opponent. This demonstrates that simple, limited TRAMs applicable to a wide variety of problems still have the power to invent new solutions to problems.

Second, it is interesting to note that MINSTREL has invented two different methods of suicide from the same episodic memory. Using TRAM:Generalize-Constrain, MINSTREL transformed the “Knight Fight” episode into a plan in which a knight fights himself. Using TRAM:Intention-Switch and TRAM:Similar-Outcomes-Partial-Change, MINSTREL transforms the same episode into a plan in which a knight purposely loses a fight to a dangerous opponent. The ability to invent several solutions from a single episode shows the flexibility and power of MINSTREL’s creativity process.

Third, unlike TRAM:Similar-Outcomes-Implicit and TRAM:Similar-Outcomes-Partial-Change, TRAM:Intention-Switch will never make a creativity error. Any action which can be done intentionally can be done unintentionally, and vice versa. Unlike the Similar-Outcome TRAMs, which extrapolate the problem solver’s knowledge into new areas, TRAM:Intention-Switch redirects the problem solver into a little-used area of his knowledge. Intentional doing things accidentally (a seemingly self-contradictory idea) is a reasoning strategy that is seldom useful to a problem solver, so episodic memory is unlikely to contain general plans of this sort. Instead, TRAM:Intention-Switch redirects the problem solver to this strategy for the restricted type of problems in which it might be useful.

2.11.2 Storytelling Example

We now look at how MINSTREL’s creativity functions in the context of a larger task: storytelling.

To tell stories, MINSTREL must select a theme, instantiate the events of the theme (the plot), assure that the events of the story are consistent, achieve literary goals such as building suspense, and so on. Many of these tasks can be achieved without creativity. MINSTREL knows that knights ride horses, and hence doesn’t have to invent a way for knights to travel from place to place. But sometimes MINSTREL encounters a new problem in the course of storytelling, or becomes bored with a particular story development. In these cases, creative problem solving is used to invent a solution.

This example presents a specific task MINSTREL encountered in telling The Vengeful Princess:

The Vengeful Princess

Once upon a time there was a lady of the court named Jennifer. Jennifer loved a knight named Grunfeld. Grunfeld loved Jennifer.

Grunfeld wanted revenge on a lady of the court named Darlene because she had the berries which she picked in the woods and Jennifer wanted to have the berries. Jennifer wanted to scare Darlene. Jennifer wanted to appear to be a dragon so that a dragon would move towards Darlene. Jennifer drank a magic potion. Jennifer transformed into a dragon. A dragon move towards Darlene. A dragon was near Darlene.

Grunfeld wanted to impress the king. Grunfeld wanted to move towards the woods so that he would fight a dragon. Grunfeld moved towards the woods. Grunfeld was near the woods. Grunfeld fought a dragon. The dragon died. The dragon was Jennifer. Jennifer wanted to live. Jennifer tried to drink a magic potion but failed. Grunfeld was filled with grief.

Jennifer was buried in the woods. Grunfeld became a hermit.

MORAL: Deception is a weapon difficult to aim.

Titles for MINSTREL’s stories were provided by the author. Throughout this dissertation, MINSTREL’s stories are presented exactly as produced, except for typography.
The particular portion of *The Vengeful Princess* this example focuses on is the creation of Jennifer's reason for wanting revenge on Darlene:

... Jennifer wanted revenge on a lady of the court named Darlene because Darlene had the berries which she picked in the woods and Jennifer wanted to have the berries.

When telling this story, MINSTREL knew nothing about what kinds of goal conflicts might lead a lady of the court to want revenge on another lady. This example shows how MINSTREL invents a conflict over possession of berries as a reason for wanting revenge.

2.11.2.1 The Problem

This example looks at how MINSTREL invents a reason for Jennifer to want revenge on Darlene. MINSTREL's representation of this problem is shown in Figure 2.18. Jennifer's goal of wanting revenge (&GOAL.1751) is motivated by a state of the world (&STATE.992) that achieves Darlene's goal (&GOAL.3029) at the expense of Jennifer's goal (&GOAL.2112). (One of the interesting storytelling aspects of this story is that MINSTREL knows that Jennifer wants revenge before it knows why Jennifer wants revenge. This is a consequence of how MINSTREL develops stories from the theme outward.)

When this example begins, MINSTREL has three goals: (1) to instantiate the state that causes Jennifer to want revenge (&STATE.992), (2) to instantiate Darlene's goal achieved by this state (&GOAL.3029), and (3) to instantiate Jennifer's thwarted goal (&GOAL.2112).

2.11.2.2 Episodic Memory

For storytelling, MINSTREL's episodic memory contains 10 story fragments from the King Arthur domain. For this example, the only relevant episode is "Picking Berries":

**Picking Berries**

A lady named Guinevere who wanted berries went to the woods and picked some.

2.11.2.3 Standard Problem Solving During Storytelling

MINSTREL uses creative problem solving to instantiate Jennifer's thwarted goal (&GOAL.2112 in Figure 2.18). But before MINSTREL does this, it instantiates the state which thwarts this goal (&STATE.992 in Figure 2.18). To instantiate this state, MINSTREL uses standard case-based problem solving.

TRAM:Standard-Problem Solving is a TRAM that implements standard case-based problem solving. Given a problem description, TRAM:Standard-Problem Solving tries to recall from episodic memory an exactly similar problem. If it can, it uses the solution from that previous problem to solve the current problem. TRAM:Standard-Problem Solving is illustrated in Figure 2.5.

In this case, the problem description is &STATE.992: "Something happens which fulfills a princess's goal." Without transformation, this recalls a similar episode from memory: the "Picking Berries" story fragment. This scene is used to fill in as much of &STATE.992 and surrounding schemas as possible. The scene development at this point is shown in Figure 2.19. Notice that this has also resulted in the addition of a new schema to the story, &ACT.1178. This new schema represents Darlene's action in picking the berries.
This example illustrates how standard case-based problem solving occurs in MINSTREL. TRAM:Standard-Problem Solving recalls an episode from memory and applies it without modification to the current problem. What happens when TRAM:Standard-Problem Solving cannot recall an appropriate episode?

2.11.2.4 Creative Problem Solving During Storytelling

MINSTREL now tries to instantiate Jennifer’s thwarted goal. To do this, MINSTREL must show how Darlene’s possession of the berries could thwart one of Jennifer’s goals. MINSTREL’s author-level representation of this goal is &GOAL.3067, and Figure 2.20 shows a trace of MINSTREL achieving this goal.

The first part of this trace shows MINSTREL recalling author-level plans for instantiating a story scene. MINSTREL recalls 4 plans. The first, ALP:Don’t-Instantiate, fails. The second, ALP:General-Instantiate, succeeds. ALP:General-Instantiate tries to instantiate a story scene by using creative problem solving at the character level. The second portion of the trace (beginning with “TRAM Cycle: &GOAL.2112”) shows creative problem solving being used to instantiate the scene.

ALP:General-Instantiate passes the story scene to be instantiated to problem solving. TRAM:Standard-Problem Solving is tried but fails, because MINSTREL does not have any scenes in episodic memory in which a lady’s goal is thwarted by someone else possessing some berries. (In fact, MINSTREL’s memory does not contain any scenes in which a lady’s goal is thwarted.)

TRAM:Standard-Problem Solving fails, so it is discarded. The next TRAM used is TRAM:Opposite-State-Achieves. This TRAM suggests that the opposite of a state that achieves a goal will thwart the goal, and vice versa. If being healthy achieves the goal of protecting your health, then being dead will likely thwart the goal of protecting your health, and so on. So to invent a thwarted goal, you can recall an achieved goal and reverse it.

In this case, the opposite of the thwarting state (Darlene’s possession of the berries) is Darlene not possessing the berries (i.e., someone else possessing the berries). TRAM:Opposite-State-Achieves changes the problem specification from “A princess’s goal is thwarted by Darlene possessing some berries” to “A princess’s goal is achieved by someone possessing some berries.” If TRAM:Opposite-State-Achieves can recall something similar to this new specification, it can be used in the original problem by reversing the recalled scene from achievement to thwarting.

MINSTREL constructs this opposite state and tries to recall goals from episodic memory that are achieved by this new state. This recalls &GOAL-BERRIES, which is Guinevere’s goal of wanting to possess berries from the “Picking Berries” episode: “Guinevere’s goal of possessing berries is achieved by Guinevere possessing the berries.”

The recalled episode can be adapted to the original problem by filling Jennifer and Darlene into the correct roles and “reversing” the outcome. This is achieved in three steps: (1) replacing the actor of the goal with Jennifer (i.e., “Jennifer’s goal of possessing the berries is achieved by Guinevere possessing

Figure 2.20 MINSTREL Trace of Story-Level Creativity


(JENNIFER WANTED TO HAVE THE BERRIES *PERIOD*)

thwarted by someone else possessing some berries. (In fact, MINSTREL's memory does not contain any scenes in which a lady's goal is thwarted.)
the berries"), (2) replacing the possessor of the berries with Darlene (i.e., "Jennifer's goal of possessing the berries is achieved by Darlene possessing the berries"), and (3) by replacing the achievement with thwarting (i.e., "Jennifer's goal of possessing the berries is thwarted by Darlene possessing the berries"").

The adapted solution can now be used to fill in the original scene. The result is shown in Figure 2.21. In English, the scene shown in Figure 2.21 is expressed:

... Jennifer wanted revenge on a lady of the court named Darlene because Darlene had the berries which she picked in the woods and Jennifer wanted to have the berries.

Notice what has happened during the creation of this thwarted goal. Prior to creating this goal, MINSTREL had no explicit knowledge about conflicts of possession, or of the idea that one person's possession of an object prevents another person from also possessing the object. By using a very simple story episode and a general creativity heuristic, MINSTREL was able to invent these concepts and apply them to a specific problem. And as these concepts are invented they are indexed into episodic memory, where they are available for future problem solving, or as a basis for additional creativity. In this way, MINSTREL uses creativity to constantly expand its knowledge, and avoids having to reinvent the same concepts over and over.

2.11.2.5 Summary

This example demonstrates MINSTREL's use of problem solving and creativity during storytelling. By using TRAM:Opposite-State-Achieves, MINSTREL solves a problem it cannot solve during standard problem solving. In solving this goal, MINSTREL discovers the idea of conflict over possession of an object. This is a concept not explicitly known to MINSTREL before its invention in The Vengeful Princess, and illustrates how MINSTREL's creativity can extend its knowledge.

2.12 Issues in Creativity

This chapter has no doubt raised in the reader's mind a number of issues about creativity and MINSTREL's model of the creative process. This section addresses the more common questions about MINSTREL's creativity model.

2.12.1 Errors in Creativity

The previous section showed how MINSTREL used creativity to invent a scene in which Jennifer wants revenge on Darlene because Darlene has some berries that Jennifer wants. Readers of this scene sometimes complain that possession of berries is insufficient motive for revenge. MINSTREL's creative problem solving has invented an incorrect solution, at least according to some readers. How did this happen, and what does it say about the creative process?

When MINSTREL begins telling The Lady's Revenge, it has only minimal knowledge about revenge. MINSTREL knows that revenge can be motivated by a thwarted goal, but it does not have any specific examples of revenge in the King Arthur domain in memory, and neither does it have any examples of thwarted goals in the King Arthur domain. Creativity must be used to extend MINSTREL's knowledge about both thwarted goals and revenge.

MINSTREL's creativity heuristics correctly extend its knowledge about possession of objects and thwarted goals. The scene in which Darlene takes the berries that Jennifer wants was created by TRAM:Opposite-State-Achieves from a scene that contained no thwarted goals. Initially MINSTREL knows that possessing an object can achieve a person's goal of controlling an object. After telling The Lady's Revenge, MINSTREL has discovered that possessing an object can also thwart another person's goal of controlling an object.

The same creativity heuristic is less successful in extending MINSTREL's knowledge about revenge. The thwarted possession of berries is invented as motivation for a revenge goal, but as most people recognize, that is probably insufficient motivation. Revenge is only plausible if the retribution is commensurate with the offense, or if the character seeking revenge is evil and likely to seek revenge out of proportion with the offense.
Creativity extends a problem solver’s knowledge. By making good use of what he already knows, a problem solver can often make accurate guesses about what he doesn’t know. But sometimes he’ll be wrong. That is the nature of creativity. Unlike first-order predicate logic and standard problem solving, creativity is a heuristic activity that trades infallibility for power. Creativity is able to discover many solutions that standard problem solving cannot because it takes risks. In this case, MINSTREL makes a reasonable guess about how the world operates—that possession of an object prevents possession by another, and that thwarting a possession goal is reason for revenge—but the guess is not completely correct.

Of course, this particular error could be easily corrected. MINSTREL could be given additional knowledge about revenge or possession of objects, and hence avoid this error. As this points out, creativity alone cannot extend a problem solver’s knowledge indefinitely or without error. The problem solver must also be able to learn directly from outside experience. In a complete cognitive model, creativity must act in concert with noncreative learning to correctly extend a problem solver’s knowledge of the world.

But no matter how much knowledge a problem solver has, creativity will extend beyond the borders of that knowledge, and errors of this type will continue to occur. The challenge is to find a model of creativity that will limit errors while still providing the power to discover novel, useful ideas.

MINSTREL’s model of creativity limits errors by using creativity heuristics that make specific, small problem transformations. By limiting the changes made to a problem specification, MINSTREL increases the likelihood that any solution it discovers will apply correctly to the original problem. The success of this strategy is apparent even in MINSTREL’s failures. “Possessing berries” may be a poor reason for revenge, but not an unreasonable one. The reader may judge it insufficient motivation, but he does at least understand the reasoning.

It is worth noting that in a complete cognitive system, errors in creativity would lead to learning experiences. If MINSTREL were able to learn from criticism, the reader’s comments about the possession of berries being insufficient motivation for revenge would be an opportunity for MINSTREL to refine its knowledge about revenge and possession of objects, and avoid similar mistakes in the future. Creativity can thus be seen as a motivator and director of noncreative learning. At this time, however, MINSTREL has no ability to learn from criticism.

### 2.12.2 Learning in MINSTREL

Although MINSTREL does not have the ability to learn from criticism, it is able to do a simple sort of learning from creativity. When MINSTREL invents a new solution to a problem, the solution is indexed in episodic memory according to the problem it solves. So when MINSTREL discovers that a conflict over possession of berries is a reason for revenge, that new knowledge is stored in episodic memory, where it can be used in future problem solving. This serves several purposes.

First, it permanently extends MINSTREL’s knowledge. If MINSTREL did not remember its past inventions, it would always begin its creative problem solving from the same base of knowledge, and would always explore the same area around the edges of its knowledge. But if a useful new solution is discovered and incorporated into MINSTREL’s knowledge base, the new solution becomes an island from which MINSTREL can continue its creative exploration. By saving its creative successes, MINSTREL increases its ability to discover new solutions. The larger MINSTREL’s episodic memory, the vaster its experience, the more powerful and effective its creativity.

Second, saving successful new solutions improves MINSTREL’s efficiency as a problem solver. Standard case-based problem solving is very efficient, because it finds solutions that apply immediately and with a minimum of effort. Creativity is less efficient, because it must search the problem space and apply solution adaptations. By remembering past solutions, MINSTREL avoids having to reinvent them, and this improves MINSTREL’s efficiency.

Both these benefits require MINSTREL to save successful solutions. Saving solutions that are wrong or even doubtful (like the “possession of berries” solution) can be counterproductive, because it will lead to MINSTREL repeating its past mistakes. It should be obvious, then, that criticism is an important element to a creative problem solver that learns from its creativity. Although MINSTREL does not currently address this issue, it is an area for future research.

Experiments that study MINSTREL’s ability to learn from creativity and the effect that has on MINSTREL’s problem solving behavior are presented in Chapter 8.

### 2.12.3 MINSTREL’s Efficiency

Saving invented solutions improves MINSTREL’s efficiency in the long run because it allows standard problem solving to solve problems that would otherwise require creativity. But how can we characterize the efficiency of creativity in the short run, that is, for a particular problem solving effort?

To begin with, creativity does not become inefficient as episodic memory grows. Retrieval from episodic memory is proportional to the number of significant features in the problem description, not upon the number of episodes in
memory. Episodic memory is organized as a tree based on the values of significant features of the indexed episodes (Kolodner, 1984; Reiser, 1986; Schank, 1982). Retrieval involves comparing the significant features of the recall description with the branches of this tree (i.e., traversing a multibranched tree). Since there is one comparison for each feature of the recall description, the time efficiency of recall is characterized by the number of features.

In fact, creativity tends to become more efficient as episodic memory grows. As memory grows, the likelihood that a transformed problem description will recall a solution increases, and so the likelihood of creativity succeeding also increases. There is no easy way to characterize this trend, because creativity does not search memory in an orderly fashion, and memory does not grow in an orderly fashion. But in general, the more the knowledge captured in episodic memory, the more likely it is that creativity will discover a solution. Some studies of how MINSTREL's behavior changes as episodic memory changes are discussed in Chapter 8.

The efficiency of creativity depends primarily upon the number of creativity heuristics that are applicable to a particular problem. If a creative problem solver has three heuristics that apply to a problem, and tries every combination of these heuristics without finding a solution, the problem solver will try 24 different combinations. In the worst case (when no solution is found), creativity is O(n + 1)), where n is the number of applicable heuristics.

In practice, a creative problem solver is likely to limit the amount of time and effort expended to find a solution. MINSTREL, for example, applies no more than three heuristics simultaneously, regardless of how many are applicable. Because combinations of heuristics search the problem space farther and farther from the original problem specification, they are correspondingly less likely to discover a solution (although if they do it is likely to be quite different from known solutions). Consequently, the problem solver soon reaches a point of diminishing return and abandons the search for a solution.

Experience with MINSTREL also indicates that because MINSTREL's creativity heuristics are very specific only a few of the available heuristics apply to any given problem. An analysis of the problems solved by MINSTREL in telling stories based on four different story themes revealed that of MINSTREL's 24 creativity heuristics, on average only 1.4 TRAMs applied to any particular problem. This indicates that MINSTREL's strategy of specific problem transformations not only increases the likelihood of discovering a useful solution, it also limits the effort expended in searching for a solution. (For experiments and studies on MINSTREL's usage of TRAMs, see Chapter 8).

---

8 We assume a constant-time hashing function to traverse the memory tree.
3
A Model of Storytelling

3.1 Why Tell Stories?

Authors create stories for a bewildering variety of reasons. A mother spins a fanciful bedtime tale to lull her young child asleep; a rabbi crafts an elegant anecdote to illustrate the generosity of God; a distressed young woman writes a novel to heal the grief she feels over losing her mother. And not only do authors write for many different reasons, they often pursue many goals at once in their writing. Shakespeare wrote works that both illuminate the human condition and delight the ear; Jonathan Swift wrote stories that were both entertaining adventures and biting social commentary on the England of his day. The goals and purposes of storytelling are as diverse and varied as human intellect itself.

But whatever the reasons, it is clear that human authors write intentionally. The stories authors create are carefully crafted to achieve particular goals. These goals and the ways they are achieved differ greatly from author to author, but every author has an explicit awareness of writing as a way to achieve some personal goals.

The importance of author goals in storytelling is best illustrated by an early model of computer storytelling called TALESPIN. TALESPIN was a computer program developed at Yale by James Meehan (Meehan, 1976). TALESPIN had knowledge about the likely goals and plans of a cast of simple woodland creatures. To tell a story, TALESPIN generated some likely goals for these creatures and then simulated their attempts to achieve those goals:
John Bear is somewhat hungry. John Bear wants to get some berries. John Bear wants to get near the blueberries. John Bear walks from a cave entrance to the bush by going through a pass through a valley through a meadow. John Bear takes the blueberries. John Bear eats the blueberries. The blueberries are gone. John Bear is not very hungry.

As this example illustrates, TALESPIN often told stories that lacked purpose. The characters act in reasonable ways and the story world is consistent and detailed, but the stories have no point or reason. TALESPIN's stories don't read like stories.

The reason for this is simple. TALESPIN knows about the characters in its story world, about the kinds of things they can do and the kinds of goals they can have, but TALESPIN lacks any knowledge about itself as an author. TALESPIN does not know why it tells stories. At best TALESPIN has an implicit understanding of storytelling as "making characters do something to achieve likely goals." But because TALESPIN focuses on character-level goals rather than author-level goals, its characters have the purpose its stories lack.

Clearly storytelling is more than creating plausible accounts of how characters might achieve their goals. Authors are not purely "simulators" of reality; they have purpose and intention in their writing. The events of a story are crafted to fulfill goals other than a mere slavish consistency with real life. To be cognitively plausible, and to create stories with purpose and direction, a model of storytelling must explicitly represent the author's goals and the process of achieving those goals.

Stories are the purposeful achievement of author goals.

Aside from its ability to be creative, MINSTREL's fundamental advancement over TALESPIN is an explicit author model. Like a human author, MINSTREL tells stories to achieve particular goals. As MINSTREL tells a story, it has an agenda of author-level goals it is trying to fulfill, such as illustrating a specific story theme, and building suspense in a particular part of the story. Because MINSTREL is a purposeful storyteller with knowledge of its goals as an author, it creates stories that are better organized, more purposeful, and more recognizable as "stories" than those created by TALESPIN.

3.2 Author Goals

If the uses of storytelling are as diverse as human intellect itself, then surely cataloging the goals of authors is a hopeless task. How then can we learn about the authoring process?

One way to begin is by identifying and defining the goals that are necessary to tell stories of a particular type. The hope is that by carefully examining the authoring process for one particular type of writing, something will be discovered about the authoring process in general. So although we may not understand everything about why authors write, we will learn something about how they write. And this knowledge will serve as a basis for further research that will lead to a deeper and more general understanding of author-level goals.

This approach has led to MINSTREL, a computer program that tells short, theme-based stories about King Arthur and his Knights of the Round Table. Narrowing the range of storytelling to a specific style, a specific length, and a specific milieu makes the storytelling problem manageable and permits MINSTREL to focus on the process of storytelling rather than the diverse "whys" of storytelling.

Restricting MINSTREL to theme-based stories limits the types of author goals MINSTREL must solve. Selecting a single, specific primary author goal—to tell a story that illustrates a particular theme—greatly narrows the range of author goals. MINSTREL does not have to tell bedtime stories, satires, or any of the other myriad types of stories. At the same time, theme-based stories are complex and rich enough to address a variety of issues in storytelling, the way certain storytelling styles—such as the stories of very young children, or mathematical story problems—would not.

Restricting the length of the stories it tells to about one page allows MINSTREL to concentrate on stories in which immediate character actions predominate. Longer works such as novels often use character interactions, interplays of moods and emotions, digressions, and complicated presentation techniques to effect their purposes. Limiting the length of MINSTREL's stories concentrates this research on how one particular tool—creating story events—can be used to achieve a variety of author-level goals.

Finally, restricting MINSTREL's storytelling to a specific milieu focuses this research on issues in storytelling rather than issues in understanding and representing knowledge about the world. The King Arthur milieu is relatively straightforward: knights love princesses and kill dragons, hermits live in the woods and heal people. Were MINSTREL to tell stories in a more complicated milieu, or in several different milieus, more effort would have had to be expended to give MINSTREL knowledge about those milieus. Although this might have led to some interesting results, the time and effort it would take to understand and represent knowledge about different milieus would have subtracted from the time available to develop a general model of storytelling and
creativity. It was decided early in this research effort that the development of
the general models of storytelling and creativity was of greater interest, and limit-
ing MINSTREL to a single storytelling domain permitted a more in-depth
development of this area of the storytelling model.

3.2.1 MINSTREL’s Author-Level Goals

Limiting MINSTREL to telling short, theme-based stories about King Arthur
revealed four important classes of author-level goals:

1. Thematic Goals
2. Drama Goals
3. Consistency Goals
4. Presentation Goals

Thematic goals are concerned with the selection and development of a story
theme. Drama goals are concerned with the use of dramatic writing techniques
to improve the artistic quality of a story. Consistency goals focus on creating a
story that is plausible and believable. And presentation goals are concerned with
how a story is presented to the reader.

To further explain these goals and illustrate how they combine to create a complete
story, we look at the role each class of goals play in one of MIN-
STREL’s stories. The story we use is called Richard and Lancelot. Except for
typography, it is reproduced here exactly as written by MINSTREL:

Richard and Lancelot

It was the spring of 1089, and a knight named Lancelot returned to Camelot from elsewhere.
Lancelot was hot tempered. Once, Lancelot lost a joust. Because he was hot tempered, Lancelot wanted
to destroy his sword. Lancelot struck his sword. His sword was destroyed.

One day, a lady of the court named Andrea wanted to have some berries. Andrea went to the woods.
Andrea had some berries because Andrea picked some berries. At the same time, Lancelot’s horse moved
Lancelot to the woods. This unexpectedly caused him to be near Andrea. Because Lancelot was near
Andrea, Lancelot saw Andrea. Lancelot loved Andrea.

Some time later, Lancelot’s horse moved Lancelot to the woods unintentionally, again causing him to
be near Andrea. Lancelot knew that Andrea kissed with a knight named Frederick because Lancelot saw
that Andrea kissed with Frederick. Lancelot

believed that Andrea loved Frederick. Lancelot loved Andrea. Because Lancelot loved Andrea,
Lancelot hated Frederick. Because Lancelot was hot tempered, Lancelot wanted to kill Frederick. Lancelot went to
Frederick. Lancelot fought with Frederick. Frederick was dead.

Andrea went to Frederick. Andrea told Lancelot
that Andrea was siblings with Frederick. Lancelot
believed that Andrea was siblings with Frederick.
Lancelot wanted to take back that he wanted to kill
Frederick, but he could not because Frederick was dead. Lancelot hated himself. Lancelot became a
hermit. Frederick was buried in the woods. Andrea
became a nun.

Moral: ‘‘Done in haste is done forever.’’

This story was selected to illustrate MINSTREL’s storytelling goals because,
more so than MINSTREL’s other stories, this story is straightforward and obvi-
ous in achieving those goals. This is particularly apparent in the third paragraph,
where MINSTREL produces a long, overly detailed explanation of the character
actions and reasoning. But although this makes the story clumsy and somewhat
difficult to read, it also makes it easier to follow MINSTREL’s purposes in cre-
ating the story.

3.2.2 Thematic Goals

The theme of a story is the main point or purpose of the story. Because there are
many possible reasons to tell a story there are many possible story themes.
MINSTREL tells stories about a particular type of theme called a Planning
Advice Theme, or PAT. Planning Advice Themes represent concise pieces of
advice about planning, and they can often be summarized by adages, such as “A
bird in the hand is worth two in the bush.”

MINSTREL’s author-level thematic goals are concerned with selecting and
illustrating a story theme. Richard and Lancelot is based on a Planning Advice
Theme called PAT:Hasty-Impulse-Regretted. This theme advises a planner to
avoid making hasty decisions that cannot be retracted if they turn out to be
incorrect. The events in a story that illustrate the theme are called the story plot.
In Richard and Lancelot, the following scenes illustrate the theme of the story:

Lancelot knew that Andrea kissed with a knight named Frederick because Lancelot saw that Andrea kissed
with Frederick. Lancelot believed that Andrea loved
Frederick. Lancelot loved Andrea. Because Lancelot loved Andrea, Lancelot wanted to be the love of Andrea. But he could not because Andrea loved Frederick. Because Lancelot was hot tempered, Lancelot wanted to kill Frederick. Lancelot went to Frederick. Lancelot fought with Frederick. Frederick was dead.

Andrea told Lancelot that Andrea was siblings with Frederick. Lancelot believed that Andrea was siblings with Frederick. Lancelot wanted to take back that he wanted to kill Frederick, but he could not because Frederick was dead.

These events form an example of the abstract advice represented in the story theme. By structuring the stories it tells around themes, MINSTREL assures that they will have the purpose that was missing from stories told by TAESPIN.

MINSTREL has two author-level thematic goals. The first goal is to select a theme for storytelling. The second is to create a sequence of story events that form an example of the selected theme.

Chapter 4 discusses MINSTREL’s representation of story themes, MINSTREL’s thematic goals, and the plans MINSTREL uses to achieve those goals.

3.2.3 Drama Goals

Human authors use a wide variety of techniques to improve the craftsmanship and literary quality of their stories. Foreshadowing, characterization, irony, suspense, and tragedy are all examples of writing techniques that authors use to improve the quality and impact of their stories. Using these techniques is rarely the primary purpose of an author’s storytelling. Instead, these are secondary writing goals that improve the artistic values of a story while supporting the theme of the story.

MINSTREL implements four drama goals: suspense, tragedy, foreshadowing, and characterization. Two of these techniques are used in Richard and Lancelot.

Foreshadowing is used to increase the impact of the scene in which Lancelot (erroneously) discovers that Andrea loves another knight by echoing parts of that scene earlier in the story:

At the same time, Lancelot’s horse moved Lancelot to the woods. This unexpectedly caused him to be near Andrea. Because Lancelot was near Andrea, Lancelot saw Andrea. Lancelot loved Andrea.

...[more text]

Some time later, Lancelot’s horse moved Lancelot to the woods unintentionally, again causing him to be near Andrea. Lancelot knew that Andrea kissed with a knight named Frederick because Lancelot saw that Andrea kissed with Frederick.

Lancelot’s willful horse first causes him to unexpectedly meet and fall in love with Andrea, and then later causes him to unexpectedly see Andrea kissing Frederick and fall out of love with Andrea. This juxtaposition and repetition of similar scene elements improves the impact of the story theme by echoing a strengthening the underlying pattern of the story.

Characterization is used to establish the hot temper of Lancelot, which contributes to his later hasty decision:

Lancelot was hot tempered. Once, Lancelot lost a joust. Because he was hot tempered, Lancelot wanted to destroy his sword. Lancelot struck his sword. His sword was destroyed.

To develop the characterization of Lancelot as hot-tempered, MINSTREL creates a story scene which shows how his hot temper affects how he reacts to events. By establishing the personality of the main character early in the story, MINSTREL improves the plausibility of later events and enhances the overall quality of the story. Chapter 5 discusses MINSTREL’s use of dramatic writing techniques.

3.2.4 Consistency Goals

Another concern for authors is to tell stories that are consistent and believable. Characters should act rationally and events should happen in accordance with the author’s best understanding of how the world functions. Readers expect stories to reflect and agree with what they know about the world, and so the author must take care to maintain that plausibility, and to explain it when absent or different from common understanding.

Story inconsistencies normally arise as side-effects of other author-level goals. For example, when MINSTREL creates the story events needed to illustrate a story theme, it creates only the events necessary for the theme. This might include a scene in which a character dies. But unless the theme happens to also include scenes explaining how the character died, who killed him, and what emotional reactions all the characters in the story had to the character’s death, the resulting story will be incomplete. The reader expects explanations of how and why things happen. The purpose of MINSTREL’s consistency goals is to detect these types of situations and to correct them by adding explanatory story events.
MINSTREL implements a variety of author-level goals aimed at maintaining story consistency. One class of goals checks to see that characters are shown achieving all the steps of successful plans:

Lancelot was hot tempered, Lancelot wanted to kill Frederick. \textit{Lancelot went to Frederick.} Lancelot fought with Frederick. Frederick was dead.

In \textit{Richard and Lancelot}, Lancelot's murder of Frederick was created to illustrate the story theme. After this scene is created, a consistency goal notices that a necessary precondition to fighting someone—being colocated with them—hasn't been fulfilled. Although Frederick's death is necessary to illustrate the story theme, an explanation of how Lancelot and Frederick came to be in the same place is not, and so MINSTREL's thematic goals did not create one. To make the story understandable, a story event is created that achieves the colocaton precondition. Consistency goals "repair" the story by noticing and correcting inconsistencies left over from other author-level goals.

Another class of goals checks to be sure that characters are reacting properly to events in their world:

Lancelot wanted to take back that he wanted to kill Frederick, but he could not because Frederick was dead. \textit{Lancelot hated himself.}

People normally have emotional reactions to the events in their lives. They feel happy when they achieve important goals, sad when a major plan fails, anxious when they are worried about their self-preservation, and so on. In this example, Lancelot discovers that he has violated a major goal because of a character flaw, but has no emotional reaction to this event. A consistency goal notices this and creates a scene describing a plausible emotional reaction. This improves the consistency of the story and the believability of the character.

The range of MINSTREL's consistency goals and the plans MINSTREL uses to achieve them are discussed in Chapter 6.

3.2.5 Presentation Goals

Presentation goals concern how the story is communicated to the reader. The author of the story must decide the order in which events in the story are presented to the reader, which events must be fully described and which can be summarized or omitted, and how each story event will be expressed in English.

\textit{Richard and Lancelot} contains the following sequence of story scenes:

Lancelot was hot tempered. Once, Lancelot lost a joust. Because he was hot tempered, Lancelot wanted to destroy his sword. Lancelot struck his sword. His sword was destroyed...

Because Lancelot was hot tempered, Lancelot wanted to kill Frederick. Lancelot went to Frederick. Lancelot fought with Frederick. Frederick was dead...

The first scene is created by MINSTREL to illustrate the characterization of Lancelot as hot tempered. The second scene is part of the theme, and turns upon Lancelot's hot temper. MINSTREL's presentation goals must recognize the purposes of these two scenes and use that knowledge to order them correctly.

MINSTREL's presentation goals are also concerned with selecting scenes to be in the story and with expressing story events in English. For more on MINSTREL's presentation goals and how they are achieved, see Turner (1993).

3.3 Author-Level Planning and Problem Solving

The previous sections identified four classes of important author-level goals and showed how they combined to create a complete story. Now we shift our attention to the process of how those goals arise and are achieved.

As noted earlier, authors tell stories to achieve particular goals. The process of storytelling involves selecting a goal from the author's pool of goals, trying to find a plan to achieve that goal, and executing the plan, possibly adding new goals to the agenda or deleting old ones. This continues until the author is satisfied with the story, that is, until the author has no more unsatisfied author-level goals.

MINSTREL models this using two processes. The \textit{planning process} is concerned with the management of author-level goals. The planning process maintains the pool of author goals and when necessary, selects a goal to achieve. The \textit{problem solving process} is concerned with solving author-level goals. It takes a goal selected by the planning process and finds, evaluates, and executes a plan to achieve that goal. These processes are illustrated in Figure 3.1.

3.3.1 Author-Level Planning

MINSTREL uses an agenda-based planning model, patterned on similar models in Lenat (1976) and Warren (1978). As goals arise they are given a priority and placed on an agenda. Priorities are represented by integer numbers on a scale of 1 to 100. At each planning cycle, the highest priority goal is selected from the agenda and passed to the problem solving process.

MINSTREL begins storytelling with an initial goal to "tell a story." This goal breaks down into subgoals including selecting a theme, illustrating a theme, applying drama goals, checking the story for consistency, and presenting the story to the reader. At each cycle, MINSTREL selects the author-level goal with
goals that a straightforward, top-down approach would not be able to solve, but also simplifies MINSTREL's planning model by eliminating the need for a mechanism to correctly select amongst competing author-level goals.

MINSTREL's planning model, and in particular, the role that re-queuing of goals plays in MINSTREL's storytelling are further discussed in Chapter 8.

3.3.2 Author-Level Problem Solving

It's not unusual to view artists with something approaching awe. The process of making art seems very different from the kinds of prosaic tasks one tackles in day-to-day life. So few are successful at art, and what they produce is so different and interesting, that there is an automatic tendency to assume that an involves unique and special mental processes.

But although Koestler (1964), Wallas (1926), and others have argued that creative domains such as storytelling, art, and music are somehow fundamentally different from more mundane problem domains, most psychological evidence suggests the opposite (Weisberg, 1986). People solve problems in creative domains in much the same way they solve problems in more traditional problem solving domains. In art as in day-to-day life, people have goals, find or create plans to achieve those goals, apply the plans, evaluate the results, and so on. We are not used to thinking of artistic endeavors such as painting and music in terms of problem solving, but at a general process level there is little to distinguish between creating and playing a musical piece and creating and writing a thank you note.

MINSTREL's author-level model of problem solving is shown in Figure 3.2. Author-level goals are input at the left side, where they are used to recall similar past storytelling situations. The author-level plans used in these past situations are then adapted and applied to the current goal. Finally, the adapted plan is assessed to determine if it meets domain-specific considerations (i.e., if you are telling a realistic story you might reject a plan that would be acceptable for a fantasy). Although this model is being applied to author-level problems, it is the same case-based model used for all problem solving processes in MINSTREL.

The portion of this model within the dotted lines is the same model described in Chapter 2. All that has been changed is the types of goals being solved.

In MINSTREL, a single problem solving model is used for all problem domains, artistic and otherwise. MINSTREL's author-level storytelling goals are solved by the same process used to solve character-level goals in the King Arthur domain and to invent devices in the mechanical invention domain. Uniformity of the problem solving process is a fundamental tenet of this research.

The process of problem solving is invariant across problem domains.

One interesting consequence of MINSTREL's uniform model of problem solving...
is that the same creative process used to invent new story scenes and new solutions to problems in the King Arthur domain is also active in problem solving at the author level. But before we can look at the role of creativity in solving author-level goals, it is necessary to digress momentarily to discuss the representation of author-level goals and plans.

3.3.2.1 Representation of Author-Level Goals And Plans

Character-level goals in MINSTREL's stories are represented using goal schemas. For example, Lancelot's goal to kill a dragon is represented as a &GOAL schema with appropriate values for the TYPE, ACTOR, and OBJECT slots. An example of a character-level goal is shown in Figure 3.3. MINSTREL's author-level goals are also represented using goal schemas. Each of MINSTREL's goals in telling a story is represented using a &GOAL schema and appropriate values for the TYPE, ACTOR, and OBJECT slots. Figure 3.4 shows an example of an author-level goal to check the consistency in a particular story scene. Note that the actor of this goal is &MINSTREL, a symbol that MINSTREL uses to refer to itself, and that the object of this goal is an event in the story being told. This example author-level goal represents MINSTREL's desire to check the schema pointed to by the OBJECT slot (&State.99) for consistency.

This consistent representation permits MINSTREL to treat character-level and author-level goals identically. The same processes can be used to store, recall, and manipulate both types of goals.

Unlike the representation of goals, MINSTREL's representation of plans is *not* consistent across the character and author levels. Character-level plans in MINSTREL are represented as interconnected collections of goal, act, and state schemas. Figure 3.5 shows the representation for a knight's plan to achieve the goal of destroying a dragon. This plan is represented as a goal (destroy a dragon, &Goal.15), a plan to achieve that goal (fight the dragon, &Act.17), and the result of executing that plan (the dragon is dead, &State.7).

Although the same type of representation could be used for MINSTREL's author-level plans, it would be clumsy and time consuming. Schemas for complicated computational actions such as looping, recursion, and so on would have to be defined and an interpreter built to perform those actions. Fortunately, MINSTREL is built upon a representation for computation—Lisp. Rather than reinvent the wheel, MINSTREL uses Lisp to represent its author-level plans and uses the Lisp interpreter to execute those plans.

Each of MINSTREL's author-level plans (ALPs) is a structured, independent block of Lisp code. Each ALP contains a test that determines when the plan is applicable and a body that executes the plan. Because Lisp code can be difficult to understand, the author-level plans presented in this text are shown in a structured English format. An example of this format appears in Figure 3.6. The author plan shown in Figure 3.6 is one of the plans MINSTREL uses to check the consistency of a story. This plan assures that characters who are injured react
3.3.3 Creativity in Author-Level Problem Solving

MINSTREL has a consistent representation for goals and a single model of problem solving that is used to solve problems at both the character and author levels. This model of problem solving includes creativity—the ability to invent new problem solutions when needed. Consequently, when MINSTREL cannot solve an author-level goal, creativity heuristics will be applied to try to invent a new plan for solving that goal, just as happens when MINSTREL cannot solve a character-level goal.

There are three steps to the creative problem solving process:

1. Transform the original problem specification.
2. Recall a similar past problem situation.
3. Adapt the associated plan to the original problem.

Because MINSTREL has a consistent representation for both author-level and character-level goals, and a single model of episodic memory, the first two steps of creative problem solving are the same for both author-level and character-level problem solving. The primary difference between character-level and author-level creativity lies in the third step: adapting the associated plan.

MINSTREL's character-level plans are represented by act and state schemas, and MINSTREL's creativity heuristics (TRAMs) know how to modify and adapt this representation. But MINSTREL's author-level plans are represented as structures of Lisp code, and MINSTREL's TRAMs do not know how to adapt Lisp code. MINSTREL's author-level plans are opaque and nonadaptable, and so MINSTREL cannot adapt author-level plans. This limits the creativity heuristics (TRAMs) that MINSTREL can apply when problem solving at the author level.

In case-based problem solving, plans to solve problems are found by recalling similar past problems. Creativity requires recalling plans from past problems different from the current problem and adapting them to the current problem. But because MINSTREL cannot adapt plans at the author-level (because they are opaque blocks of Lisp code) MINSTREL cannot apply any creativity heuristics which involve plan adaptation. However, there are a few types of creativity that do not require plan adaptation.

Consider, for example, a problem solver finding his way home from a newly built shopping mall. The problem solver has never come home from this mall before, so standard problem solving will not recall any ready-made plans. But if he recalls that the shopping mall stands on the site of a former restaurant he frequented, then he can recall a route for driving home from the restaurant and use...
it without change. The problem solver has invented a solution to a problem by making use of an old solution without adaptation.

We call this type of creativity "nonadaptive creativity," because it hinges upon finding a solution that can be applied to a new problem without having to change the solution.

At the character-level, we have already seen some examples of creativity heuristics that do not require adaptation. One of these is TRAM:Recall-Act. TRAM:Recall-Act suggests that if you are trying to find an act that fits some particular set of constraints, then you can probably ignore all the constraints except the goal the act is intended to fulfill and the intended effects of the action. By ignoring the other constraints, TRAM:Recall-Act permits the problem solver to recall solutions which he would not have otherwise recalled, because they would not fulfill the extraneous constraints. And because the extra problem constraints are extraneous, the recalled solution—even though it does not fill those constraints—does not need adaptation.

TRAM:Recall-Act illustrates the central feature of nonadaptive creativity: redirecting recall to a new area of memory where immediately useful plans are likely to be found. TRAM:Recall-Act finds new solutions to a problem by redirecting recall to solutions that lack the extraneous constraints.

So although MINSTREL cannot use all types of creativity at the author-level, it can use nonadaptive creativity. To understand MINSTREL's author-level nonadaptive creativity works, it is first necessary to understand how MINSTREL's author-level episodic memory is organized.

MINSTREL's author-level plans are indexed in episodic memory according to specific past goals they have solved, just as character-level plans are indexed according to past goals they have solved. So, for example, the author-level plan ALP:Make-Consistent-P-Health is indexed under a goal of type &Check-Consistency applied to a &State schema. A portion of MINSTREL's author-level episodic memory illustrating this organization is shown in Figure 3.7. Episodic memory is organized as a tree of (feature, value) pairs. &Goal.16 is an author-level goal from a past instance of storytelling in which MINSTREL's goal to check the consistency of a particular state in a story (&State.99) was achieved by ALP:Make-Consistent-P-Health. This is organized in episodic memory by the features and values of &Goal.16, including the object of the goal, &State.99. If a new goal is encountered with similar features, ALP:Make-Consistent-P-Health will be recalled.

MINSTREL also has generalized author-level plans. These are plans that can apply to a number of different objects. Generalized author-level plans have a null Object and are indexed accordingly. Figure 3.7 shows the indexing for a generalized author-level plan called ALP:Default-Consistent. ALP:Default-Consistent is a simple plan that recognizes a schema as consistent if it has had its Type slot filled in.

When MINSTREL has a goal to check the consistency of a state schema, the goal will be used as an index to memory and because its features and values match those of &GOAL.16, MINSTREL will find &GOAL.16 and its associated plan, ALP:Make-Consistent-P-Health. This is standard problem solving—recalling a similar past problem and using the solution to that problem for the current problem.

Nonadaptive creativity comes into play when this fails. When an author-level goal does not recall a similar past goal and its associated solution, or if all the
recalled solutions fail, then MINSTREL must look elsewhere in author-level episodic memory for an author-level plan that can be applied to the current goal without adaptation. There is one place that such a plan can be found: a generalized author-level plan.

Generalized author-level plans will not be found by standard problem solving because the Object slot of the current goal will not match the null slot under which the generalized plans are indexed. But precisely because these plans are generalized we know that they can be applied without adaptation to any goal of the proper type. What is needed is a creativity heuristic that will find these plans by looking in the appropriate place, that is, a nonadaptive creativity heuristic.

MINSTREL’s TRAM for achieving this is called TRAM:Generalized-AL-Plans. This TRAM finds generalized plans to apply to a specific author-level goal by eliminating the Object slot from the current goal and returning without adaptation whatever plans it finds. TRAM:Generalized-AL-Plans is shown in Figure 3.8. When TRAM:Generalized-AL-Plans is applied to the author-level goal illustrated in Figure 3.4, it eliminates the Object slot and uses the transformed goal as an index to memory. This recalls ALP:Default-Consistent, which is then applied to achieve the original goal.

Because MINSTREL’s TRAMs cannot adapt MINSTREL’s author-level plans, creativity at the author-level is limited to nonadaptive creativity. Although nonadaptive creativity is not as powerful as other types of creativity, it does illustrate the invariance of MINSTREL’s model of problem solving and demonstrate that the same creative problem solving process used to solve character-level goals can also solve author-level goals.

3.3.4 Achieving Author-Level Plans

Our examination of the authoring process began with the identification of author goals. For the particular type of stories that MINSTREL tells, we identified four important classes of author-level goals. Then we looked at the processes of planning and problem solving: How author goals were managed, and how plans to solve author-level goals were found or created. The last step is to look at the contents of author-level plans: What they do to achieve author-level goals.

The following chapters will discuss in detail each of MINSTREL’s author-level plans, identifying what goal each plan applies to, and describing how the plan achieves that type of goal. But before we turn to the details of MINSTREL’s author-level plans, we should like to examine this problem in more general terms. What does an author do to illustrate a theme, use a dramatic technique, or correct a causal inconsistency?

Consider a hypothetical author writing a short story about Lancelot, with a goal to portray Lancelot as deceptive. He achieves this goal in two steps.

First, the author uses his knowledge of the goal he is trying to achieve and his knowledge of how stories are told to specify his goal as an abstract description of story events which, if part of the story, would achieve his goal. In this case, the author knows that “deception” is a character trait and that character traits are reflected in character actions. From this he realizes that his goal of portraying Lancelot as deceptive will be achieved if the story contained some scenes in which Lancelot used a deception plan. Note that the author hasn’t yet achieved his goal. He has only further specified it as a particular, abstract description of story events which would achieve the goal.

The abstract specification the author arrives at will depend on his knowledge of storytelling. From reading and writing stories, he will have built up a library of author-level plans that translate author-level goals into story specifications. Illustrating a character trait by including a story event demonstrating the trait might be a plan the author learned through conscious study, or by reading many stories that used this technique.

Next, the author tries to create story elements to fulfill this abstract specification. Using his knowledge of the genre of the story, the goals of the story, and the already-completed portions of the story, the author tries to invent scenes to fit the abstract specification. Thinking about one person deceiving another may remind the author of a time when a coworker fooled him with a falsified memo from their boss. Being reminded of this scene, the author may decide to use it as the basis for the Lancelot story scene. But to make that reminding work in his story, the author must make some adaptations. Lancelot will have to take the coworker’s place, the memo will have to be replaced with something appropriate to the King Arthur milieu—perhaps a note from the king—and so on. The end result is a scene in which Lancelot fools Guinevere by forging a note from the King.

We call this process of taking an abstract specification and general story constraints and inventing scenes to fit the specification instantiation.

1. Define an abstract specification of a needed story element.
2. Create a specific story element to match the abstract specification.
The claim of this research is that this two-step process of specification and instantiation is the fundamental process by which author-level goals are achieved. Not all author-level goals are achieved by this process: Goals may be achieved by creating subgoals, reordering events in the story, creating English language sentences, and so on. But the process of specification and instantiation is a pervasive and fundamental element of storytelling.

3.3.5 The Role of Episodic Memory in Storytelling

How is instantiation achieved? In the foregoing example, the author instantiated his abstract specification by a process of transform, recall, and adapt, which the reader will recognize as creative case-based problem solving. In fact, instantiating an abstract concept description can be viewed as a special form of creative case-based problem solving. But instantiation differs from normal problem solving in an important way. In normal problem solving, the problem solver uses a complete problem description to recall similar past problems, so that the problem solver can use the associated plans. But in instantiation, the problem solver uses an incomplete description to recall a complete description, without any interest in the associated plans.

The product of normal problem solving is a plan for the current problem. Suppose, for example, that a knight finds himself threatened by a dragon and wants to save himself. The knight recalls a past situation in which he was threatened by a troll. He’s solved that by charging the troll on his horse, so he decides to apply that same plan to the dragon. By recalling a similar past problem solving situation, the knight has found a plan to solve his current problem.

In instantiation, though, the product of problem solving is the recalled problem situation, not the associated solution. Suppose, for example, that an author is creating a scene in which “Lancelot, a knight, is endangered.” This abstract description is passed to case-based problem solving and recalls a past situation in which a knight was threatened by a troll and consequently killed the troll by charging it on his horse. The author can now use the recalled problem situation to fill in or instantiate the current scene. The associated plan (charging the monster on horseback) may or may not be used, depending on the author’s particular storytelling needs. In fact, the scene being instantiated could be a feeling, an emotion, or some other type of story element that does not have a plan associated with it at all.

Because there is not necessarily an associated problem solution, and because instantiation may not make use of the problem solution even if it exists, there is no need to perform the second and third steps of problem solving: adapting the recalled solution and assessing the result. Instantiation requires only the recall of a similar past scene, and not the rest of the problem solving effort.

However, a problem arises when the author cannot recall a similar past scene. In this case, recall alone is not sufficient, because the author needs to create a story scene to fit his abstract scene description.

The solution is to use imaginative memory. Imaginative memory incorporates creativity into the recall process, permitting episodic memory to invent a memory to match a set of recall indices. If the recall indices are the features of an abstract scene specification, then imaginative memory will either recall or invent the specific story scene needed to instantiate the scene specification. Instantiation is thus simply the process of imagination—using creativity and the knowledge in episodic memory to imagine scenes to fit a particular criteria.

Instantiation is achieved through imaginative memory.

It’s a common intuition to link imagination with storytelling. By explicitly representing instantiation as a fundamental process in storytelling and showing how instantiation can be achieved by imaginative memory, MINSTREL identifies and defines the link between imagination and storytelling. And by showing how creativity can be incorporated into the recall process, MINSTREL demonstrates the link between creating new problem solutions and the ability to imagine plausible new situations and ideas.

3.3.6 Planning With Many Constraints

One reason storytelling is a difficult task for humans and computers alike is that it requires the storyteller to solve a large number of interdependent goals simultaneously. In a successful story, the events of a story fulfill a range of goals. They illustrate the theme of the story, develop the literary value of the story, maintain the story consistency, and so on. But it is difficult for humans to solve planning situations that involve a number of simultaneous goals (Flower & Hayes, 1980). Each added goal increases the complexity of the planning task, until it may be nearly impossible. “The act of writing is best described as the act of juggling a number of simultaneous constraints.” (Flower & Hayes, 1980, p. 31).

One solution to planning with many constraints which is often applied to the storytelling problem is to partition the problem into semi-independent subproblems (Flower & Hayes, 1980). Rather than try to solve all the problems involved in creating a story at once, the author breaks the writing process down into a sequence of goals, that is, theme, drama, consistency, presentation.

The technique of delaying constraints is often sounded in writing handbooks:...

...begin by doing some “free writing”. This simply means writing down whatever comes into your mind about a subject... this is not the time to think about spelling, punctuation, or the correct choice of words... What you’ll produce is generally called the first or “rough”
draft, and it will probably need some revisions and changes... (Tchudi & Tchudi, 1984, pp. 21-24)

Delaying constraints or partitioning the storytelling problem into semiindependent subproblems permits the author to solve problems that would otherwise be too complex. But one consequence of this strategy is that solving one subproblem may violate another. For example, MINSTREL creates the following story scene in order to illustrate the theme "Done in haste is oft regretted":

One day, Lancelot wanted to kill Frederick. Lancelot fought with Frederick. Frederick was dead.

This scene does serve to illustrate the theme (by showing Lancelot performing a hasty action he will later regret). But it is otherwise incomplete. It does not explain why Lancelot wants to kill Frederick, or how Lancelot came to be in the same location as Frederick. The goal to illustrate the theme has been achieved, but other goals have not.

Human authors have the same difficulties when they partition complex problems. Problem constraints may be violated or even forgotten. An article on revising that appeared in Writer's Digest includes a “Tactical Analysis Checklist” of mistakes of this sort:

* Look at the motives of your major story characters. Have you included sufficient information about their past lives to make it credible that they want what they want, feel what they feel, think as they think, act as they act, or have the skills they call on in your plot? (Bickham, 1992, p. 28)

For both human and computer authors, breaking a complex problem with many constraints into simpler, semiindependent subproblems is a valuable writing strategy. But because a story is an integrated whole, no subproblem can be truly independent of the other subproblems. So the author must be prepared to detect and correct constraint violations during the writing process. In MINSTREL, this is achieved through opportunistic goals.

3.3.7 Active Versus Opportunistic Goals

At any time during the storytelling process, MINSTREL has a number of goals that it is actively trying to achieve. Each of these goals is present on the goal agenda and the active goal with the highest priority is being achieved. Initially, the active goals include the goals (1) to tell a story illustrating a particular theme, and (2) to present the story to the reader in English. These goals are achieved via author-level plans that may create additional active goals (i.e., subgoals).

In addition to the active goals, storytelling conditions can cause opportunistic goals to arise. An opportunistic goal is an author-level goal that becomes active whenever a specified storytelling situation arises. For example, MINSTREL has an opportunistic goal that arises whenever a story scene is created that contains inconsistencies. When an inconsistent story scene is created an opportunistic goal to correct the inconsistency is triggered. Similarly, MINSTREL has opportunistic author-level goals to achieve various dramatic writing purposes, such as building suspense. When a story scene is created that is suitable for suspense (such as a character's life being threatened), an opportunistic goal arises to consider building suspense in that scene.

Opportunistic goals serve two purposes in MINSTREL. First, opportunistic goals provide a mechanism for detecting and correcting constraint violations. As described earlier, MINSTREL reduces the difficulty of the storytelling problem by breaking it into a sequence of semiindependent subgoals. Opportunistic goals provide a mechanism for detecting when the solution of one subgoal violates the constraints of another subgoal, as for example when a scene created to illustrate the theme of a story contains causal inconsistencies.

Second, opportunistic goals are used to represent secondary author goals. In general, we suppose that any author has a variety of goals when telling a story, some which are primary and some which are secondary. These two categories represent a sharp, qualitative divide in author priority. The author's primary goals are those which the author considers essential to the telling of the current story. If the author cannot achieve his or her primary goals, then the storytelling process has failed. For MINSTREL, the primary goals are (1) to tell a story concerning a particular theme, (2) to make the story consistent and believable, and (3) to present the story in English. If any of these goals fail, then the storytelling process itself has failed.

Secondary goals, on the other hand, represent goals that are desirable but not essential to the storytelling process. The author is pleased if he or she can achieve these goals, because it means that the story is better than it might otherwise be. But if the opportunities do not arise, the story is not a failure. MINSTREL's goals to use literary techniques are secondary goals. If they succeed, they add additional quality and complexity to the stories MINSTREL tells, but the story can succeed even if they fail. (Although it will likely have less literary value.)

Of course, how an author categorizes his or her goals varies from author to author, and from story to story. For a writer of mystery stories, building suspense is probably an essential goal in the writing process. But for the writer of romantic fantasies, it is a secondary goal, and for the writer of comedy it may be actively avoided. And even for a single author, goal priorities will vary from story to story.

In terms of the storytelling process, primary and secondary goals differ primarily in how they arise. Primary goals are self-directed, while secondary goals are reactive. In MINSTREL, primary goals arise directly from MINSTREL's
initial storytelling goal. MINSTREL's initial goal ("tell a story") creates sub-goals to tell a story about a particular theme, to tell a consistent and believable story, and to tell the story in English, that is, all of MINSTREL's primary goals. The secondary goals, on the other hand, arise in reaction to the developing story. In MINSTREL, each secondary goal is represented by an opportunistic goal. If the opportunity arises to achieve a secondary goal, then the opportunistic goal triggers. Thus primary goals are guaranteed to arise and be attempted, while secondary goals may or may not arise, depending on the story development.

It is also interesting to note that author-level plans used to achieve a primary goal are generally not suitable for achieving that goal opportunistically, and vice versa. For example, MINSTREL has an author-level plan to add suspense to a story scene by having a character attempt an escape from a dangerous situation. This plan cannot be used actively, because it depends on the prior existence of a dangerous situation. In general, plans to achieve primary goals must be able to create the story events necessary to achieve the goal from scratch, while opportunistic goals can be achieved by plans that modify or augment existing story scenes.

3.4 Conclusions

MINSTREL has three major advancements over previous models of storytelling.

First, MINSTREL demonstrates the importance of an explicit author model in storytelling. The particular author-level goals and plans MINSTREL uses to tell stories are of great interest, and are fully discussed in the following chapters. But as important is MINSTREL's architecture as a storyteller with an explicit knowledge of its own goals.

Second, MINSTREL models storytelling as problem solving. By this, MINSTREL clarifies the relationship between achievements in artistic domains such as storytelling and achievements in traditional problem solving domains such as mechanical repair. MINSTREL demonstrates the fundamental similarities between artistic endeavors and traditional problem solving, and MINSTREL is *prima facie* evidence that artistic ability can be explained in terms of problem solving, and that no further or different cognitive process need be stipulated.

Third, MINSTREL explicitly models instantiation as a fundamental storytelling process. By recognizing the importance of instantiation, MINSTREL is able to clarify and define a process that has previously been unrecognized and unilluminated. And by implementing instantiation using imaginative memory, MINSTREL further defines the links between creativity, memory, and problem solving.

4.1 Introduction

When MINSTREL tells a story, its primary goal is to illustrate a particular story theme. MINSTREL's story themes are stereotypical planning situations, which can often be summarized by an adage such as "A bird in the hand is worth two in the bush" or "Deception serves the devil." To understand the role of story themes in MINSTREL's storytelling process, there are five issues that must be addressed:

1. What is a theme?
2. How is a theme represented?
3. What themes does MINSTREL know?
4. What author-level thematic goals does MINSTREL have?
5. What plans does MINSTREL have to achieve these goals?

4.2 What Is a Theme?

Webster's New Collegiate Dictionary defines a theme as "the subject or topic of discourse or of artistic expression"; Roget's Thesaurus lists synonyms for themes that include "point," "motif," "topic," "pattern," and "design." The theme of a story is the underlying concept or topic that organizes the story into a coherent whole; it is the theme that gives rise to a story's organization and structure.

But stories are told for a wide variety of purposes. People use stories as