EM-ONE: An Architecture for Reflective Commonsense Thinking

by

Push Singh

B.S. Massachusetts Institute of Technology (1998)
M. Eng. Massachusetts Institute of Technology (1998)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Author . . . . . .
Department of Electrical Engineering and Computer Science
17 May 2005

Certified by . . . . . .
Marvin Minsky
Emeritus Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

EM-ONE: An Architecture for Reflective Commonsense Thinking

by

Push Singh

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

This thesis describes EM-ONE, an architecture for commonsense thinking capable of reflective reasoning about situations involving physical, social, and mental dimensions. EM-ONE uses as its knowledge base a library of commonsense narratives, each describing the physical, social, and mental activity that occurs during an interaction between several actors. EM-ONE reasons with these narratives by applying "mental critics," procedures that debug problems that exist in the outside world or within EM-ONE itself. Mental critics draw upon commonsense narratives to suggest courses of action, methods for deliberating about the circumstances and consequences of those actions, and—when things go wrong—ways to reflect upon and debug the activity of previously invoked mental critics. Mental critics are arranged into six layers, the reactive, deliberative, reflective, self-reflective, self-conscious, and self-ideals layers. The selection of mental critics within these six layers is itself guided by a separate collection of meta-level critics that recognize what overall problem-type presently confronts the system. EM-ONE was developed and tested within an artificial life domain where simulated robotic actors face concrete physical and social problems. A detailed scenario is presented where EM-ONE enables two such actors to work together to build a table by engaging reactive, deliberative, and reflective processes operating across the physical, social, and mental realms.

Thesis Supervisor: Marvin Minsky
Title: Emeritus Professor of Electrical Engineering and Computer Science, MIT.

Thesis Reader: Aaron Sloman
Title: Professor of Artificial Intelligence and Cognitive Science, School of Computer Science, University of Birmingham, UK.

Thesis Reader: Gerald Sussman
Title: Matsushita Professor of Electrical Engineering, MIT.

Acknowledgments

This thesis is a product of many years of thinking about the nature of the mind. I would like to thank all my friends and colleagues over the years for discussing aspects of this subject and for all their ideas and insights. In these acknowledgements I would like to specially thank those who have helped me bring this thesis to fruition.

First and foremost, I would like to thank my advisor and mentor Marvin Minsky. I cannot overstate the influence Marvin has had on my thinking. Few people have the opportunity to meet their heroes, let alone become their apprentices and eventual collaborators and good friends. In Marvin I found a mind that was truly worth understanding and emulating. I have never met anyone else so purely and powerfully dedicated to unraveling the nature of that strange and wondrous class of processes that we call *minds*, and it has been my honor to have spent these past years learning from him how to think about this difficult subject.

I would like to thank my readers Aaron Sloman and Gerry Sussman for advising me on this thesis. Aaron is the best philosopher in the field of AI, and is one of the few people in the world who has a conception of the mind's architecture that appreciates its true richness. Gerry is one of the great teachers at MIT, and cares deeply about his students—there is a good reason that every public talk Gerry gives is a special, exciting event that always packs the room. His PhD thesis on debugging was the original inspiration for this thesis.

I would like to thank the members of the Commonsense Computing Group at the Media Lab, especially Ian Eslick, Hugo Liu, and Bo Morgan. Ian Eslick was an insightful sounding board for later versions of the theory presented in this thesis, and prodded and encouraged me with just the right amount of force and at just the right intervals to get me to finish this thesis. Hugo Liu was an unceasingly creative presence, and while we did not

much discuss the content of this thesis, our discussions always inspired me to view our efforts at understanding common sense more grandly as creating the next intellectual foundation for human culture. Bo Morgan built much of the Roboverse world that I used in this thesis, and I greatly appreciate all the hard work that he put into this project and the help he has given me over the years. I would also like to thank Nick Cassimatis and Tim Chklovski, the other members of the Society of Mind Group at the Media Lab from my early days as a graduate student. Even then, when the climate for high-level AI was much worse than it is today, they were dedicated to the goal of building human-level AI, and they helped me formulate my initial thinking about commonsense reasoning. Betty Lou McClanahan administrated our group for many years, and I greatly appreciate the help and support she provided over that time.

I would like to thank the many friends, colleagues, staff, and faculty at the Media Lab who have supported me over the years. The Media Lab has been a fantastic environment for pursuing this research. Although it was difficult to do while I was finishing this thesis, when I was a younger graduate student I would regularly stroll all over the lab and talk to everyone—undergraduates, graduate students, research scientists, and faculty—about their research. Each person's interests were so completely different. The Media Lab is a kind of intellectual garden salad where very different ideas regularly collide to form ideas new to this world. I would like to extend a special thanks to Nicholas Negroponte for founding this unique place, where people have the opportunity to work on important problems that are at the intersection of existing disciplines and that differ from the prevailing winds, and to the Media Lab's current director Walter Bender for supporting and promoting my work on commonsense reasoning as an important new research direction for the Media Lab.

I would like to thank the many workers in AI inside and outside of MIT that I have spent time with over the years. Patrick Winston and his circle of students have always been an inspired and energetic group of talented individuals, fellow travelers who are truly dedicated to building a human-level intelligence. Rod Brooks helped me jump through the many hoops that MIT presented me, and is a genuine example of how to swim against

the mainstream and still succeed. Oliver Selfridge, who helped create the field of AI and whose views on adaptation and debugging influenced this thesis from its inception, has long supported me in my work, and demonstrated to me that intelligence and humanity are entirely compatible. Doug Riecken supported me at IBM for two summers when I was beginning to think about this thesis, and has been an unflinching supporter of our work on commonsense reasoning over the years. John McCarthy, even though I see him only rarely, always leaves my thinking about AI transformed after our conversations. Doug Lenat created the unique and fascinating artifact Cyc, which helped me understand what a broad coverage commonsense knowledge representation might look like, and is an example of an AI researcher whose eyes are truly on the prize. Finally, I would especially like to thank Erik Mueller for being the greatest commonsense AI hacker around. Erik has great prowess both scientific and technical, and has a profoundly deep grasp of the challenges and issues involved in building commonsense AI systems, a grasp that can only be obtained by spending many years in the trenches personally building real systems. Our many conversations about commonsense reasoning and AI in general were some of the most useful I have had with anyone.

I would like to thank the many sponsors of the Media Lab, and especially the National Science Foundation and Jeffrey Epstein, for supporting our work on commonsense reasoning. In addition to supporting our work at the Media Lab, Jeffrey Epstein also sponsored the St. Thomas Commonsense Symposium, where many of the ideas in this thesis were first publicly discussed.

The seeds of the thinking that led to this thesis go back to my childhood. I would like to thank my close friends from back home in Montreal, especially Andrew Templeton and Andy Nguyen, who even at that young age were willing to talk about advanced ideas about the mind, and my teachers, especially Dr. Cajetan Menke, for providing an environment where ideas different from the ordinary could grow.

I would like to thank my family, Mahender, Kulwant, Raminder, and Vindi, for always being busy creating new things. I lived in a constructivist household, and my concept of

what it meant to be a person was based first on their example. I would especially like to thank for father Mahender for supporting my obsessive interest in computers throughout my childhood.

Finally, I would like to thank Barbara Barry, in whom I have found a connection that spans realms of thought and feeling that I never thought were possible with a single person. She is harmony of creativity, intelligence, empathy, and understanding without whom I cannot imagine life.

Table of Contents

List of Figures

Preamble

How can we build machines with "common sense"—that is, with the thinking skills that most people share? People are capable of a wide variety of cognitive feats, including anticipating future events, inferring causes from their effects, proposing actions to move us closer to our goals, understanding the goals that motivate the actions of others, criticizing our recent thoughts in order to improve our future deliberations—and these are only a few of the mental skills we possess. Furthermore, these abilities operate across a diverse array of mental realms, such as the physical realm where we predict how objects will behave, the social realm where we reason about how to improve our relationships with others, and the mental realm where we reflect upon our own mistakes and successes. These core human competences have long remained beyond the reach of our machines. How can we give them such extraordinary powers?

It has proven difficult to build systems with much common sense. I believe this is because human commonsense thinking is a far richer phenomenon than any of the automated reasoning processes that are familiar in artificial intelligence. To illustrate this, consider the following scenario from Marvin Minsky's forthcoming book *The Emotion Machine*, illustrating the multiplicity of ways of thinking that human minds are capable of:

> *Joan is part way across the street on the way to present her finished report. While thinking about what to say at the meeting, she hears a sound and turns her head—and sees a quickly oncoming car. Uncertain whether to cross or retreat, but uneasy about arriving late, she decides to sprint across the road. She later remembers her injured knee and reflects upon her impulsive decision. "If my knee had failed, I could have been killed. Then what would my friends have thought of me?*

Minsky suggests that Joan's mind engages in many different ways of thinking during this event. Some of these produce actions in the world, others make inferences and construct

mental descriptions, and yet others are reflective, producing thoughts that are concerned not so much with the outside world, but with various aspects of Joan herself:

Reaction: She reacted rapidly to that sound.
Representation: She constructed descriptions of things and ideas.
Attention: She noticed certain things rather than others.
Decision: She selected among alternative options.
Meta-Decision: She selected some method for choosing those options.
Embodiment: She was partly aware of her body's condition.
Intention: She formulated some goals and plans.
Language: She heard words or dialogs in her mind.
Imagining: She envisioned some alternative possible futures.
Planning: She considered various action-plans.
Reasoning: She constructed various arguments.
Recollection: She constructed descriptions of past events.
Identity: She regarded herself as an entity.
Reflection: She thought about what has she recently done.
Moral Reflection: She reflected upon what she ought to have done.
Self-Reflection: She reflected on what she was recently thinking.
Self-Imaging: She engaged certain models that she's made of herself.
Social Reflection: She considered what others might think about her.
Self-Awareness: She recognized some of her mental conditions.

I believe that, like Joan, a commonsense thinking machine will need to operate on all of these levels, and that even the most ordinary problem circumstances involve many of these types of thinking.

Why does commonsense thinking need to be so complicated? Could there not be some simple, uniform strategy for reasoning and learning that could do what we do? The difficulty is that the problems we face in life are tremendously diverse in nature, so diverse than no single strategy has so far proven adequate to the task. To illustrate this, consider the situation of two children playing together with blocks shown in Figure P-1.

Figure P-1. Playing Together

Even in this simple situation, the children may have concerns that span many commonsense realms, for example the child on the left might wonder about:

- **Physical**: What if I pulled out that bottom block?
- **Social**: Should I help him with his tower or knock it down?
- **Mental**: I forgot where I left the green block.
- **Bodily**: Can I reach that green block from here?
- **Visual**: Is the green block hidden behind that stack?
- **Spatial**: Can I arrange those blocks into the shape of a table?
- **Tactile**: What would it feel like to grab five blocks at once?

No present day AI system reasons across such a broad range of realms. I believe that any commonsense reasoning system we design should aim to achieve some competence within each of these and other important realms and, like Joan, should be capable of thinking in multiple, rich ways about each of these realms.

The desire to build a system that could deliberate about commonsense realms in rich, reflective ways was what motivated the system I present in this thesis. It is based not on a single, uniform inference strategy, but instead organizes a society of "mental critics" that

embody heuristic methods for suggesting solutions to problems that exist in the world, and in the mind of the system itself.

In a sense, this thesis tries to go back to the early 1970s MIT AI Lab, whose intellectual leaders Marvin Minsky and Seymour Papert regarded thinking as the product of an elaborate, heterogeneous collection of programs, rather than as the product of a single, simple program operating on a large quantity of uniformly represented data.[1] In fact, the system described in this thesis could have straightforwardly been implemented using the technology available in those earlier years.

In this thesis I will tell a story of the thought processes that underlie a single, seemingly simple scenario where two robots work together to build a table. The reader may find that the underlying computations are more intricate than they had anticipated, and I hope they are encouraged to consider approaches to AI where the procedural side of commonsense thinking is seen as a rich and varied collection of processes and structures that deserves far more study than it has received.

---

[1] See the section entitled *Uniform procedures vs. Heuristic Knowledge* in Minsky & Papert (1972).

Chapter 1

Overview

1.1 An Architecture for Reflective Commonsense Thinking

In this thesis I will describe the EM-ONE architecture for reflective commonsense thinking. EM-ONE is capable of reasoning about commonsense scenarios involving complex interactions between several actors along physical, social, and mental dimensions. Consider the scenario depicted in the storyboard shown in Figure 1-1, in which two creatures named Green and Pink work together to build a table.

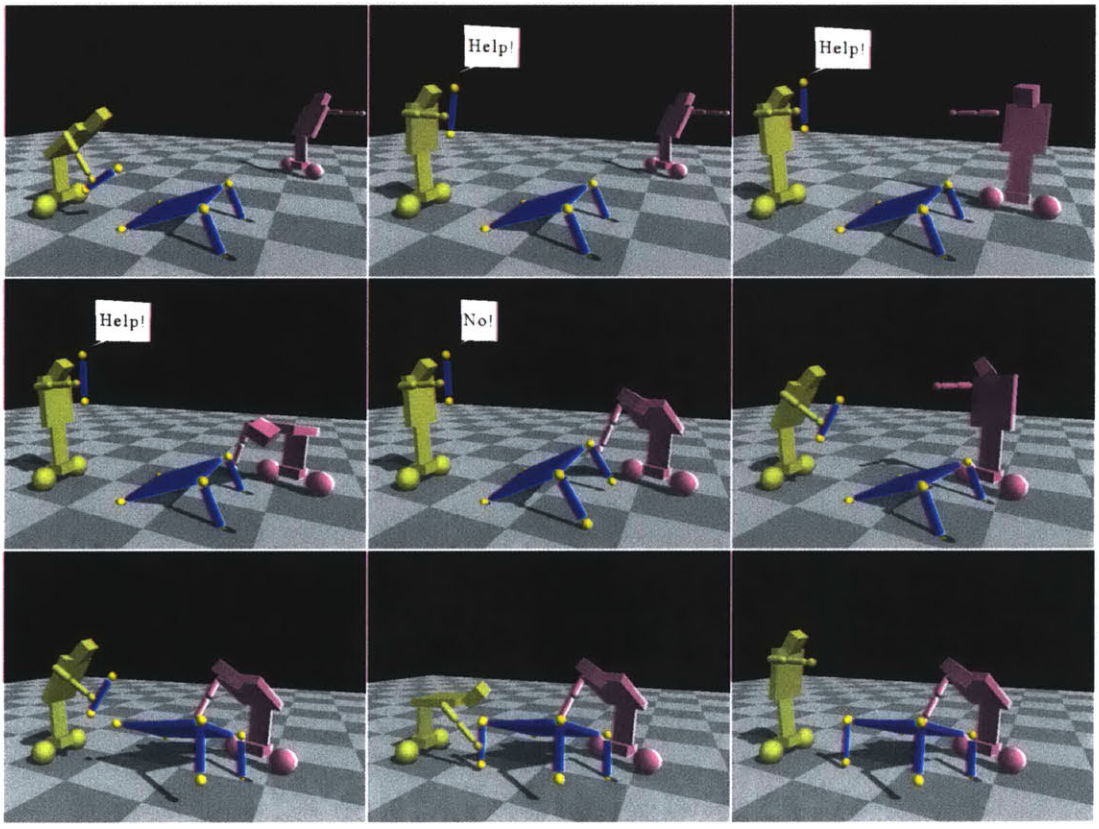


Figure 1-1. Building a table together

Here, Green wants to build a table (perhaps, to place something on.) Green sees there is already a partly built table and realizes that it needs to attach more legs to complete the table. Green goes over and grabs a stick, and then goes over to the table. Green tries to attach the stick to the table but fails. Green quickly realizes that it needs help to insert the leg under the table, because Green only has one arm. Green calls over to Pink. Pink, who has been occupied with its own projects and has not been paying attention to Green until now, looks at Green holding a stick, and infers (mistakenly) that Green is trying to disassemble the partly built table. Pink comes over and starts to detach one of the table legs. Green realizes that Pink did not correctly infer Green's intent, and so complains. Green realizes that Pink did not see Green trying to attach the table leg. Green tries to attach the stick again to the table, this time with Pink watching. Pink now realizes that Green doesn't want to disassemble the table, but rather wants to complete the table, and that Green expects Pink to hold up the table so that Green can attach the table leg it is holding. Pink holds up the table, and Green inserts the table leg underneath.

Even seemingly simple problems like this involve many kinds of cognitive processes. The above scenario requires proposing courses of actions, making inferences about the consequences of those actions and the intentions of other actors, and reflecting upon and repairing mistaken inferences, all ultimately concerned with aspects of the world that span the physical, social, and mental realms.

EM-ONE is a cognitive architecture whose purpose is to support the kinds of commonsense thinking required to produce the scenario described above. EM-ONE operates by applying *mental critics*, procedures that recognize problems in the current situation; some mental critics respond to problems in the world, and other mental critics respond to problems in the EM-ONE system itself. EM-ONE uses as its commonsense knowledge base a library of *commonsense narratives*, each a story describing a fragment of the physical, social, and mental activity that occurs during a particular interaction between two actors. Mental critics use commonsense narratives to suggest courses of action, ways to deliberate about the circumstances and consequences of those actions, and ways to reflect upon their mistakes when things go wrong.

1.2 The Critic-Selector Model

EM-ONE is based on the *critic-selector model*, a model of commonsense thinking proposed by Marvin Minsky (forthcoming) as a way to organize systems that make use of many forms of inference and knowledge representation. The central idea of the critic-selector model is that when the system encounters a problem, instead of engaging some particular general-purpose method for inference or action, it brings to bear knowledge about what AI method the system should employ to attack the problem. In other words, it thinks briefly about how it will think about the problem, and then thinks about it in that that way.

The critic-selector model operates as follows. At the top-level of the EM-ONE system are *meta-managerial critics* that react not only to the situation as it exists in the outside world, but also to internal reports about the progress that the system has made on the present problem so far. Based on that information, these critics then select a way to think about the current predicament. Examples of meta-managerial critics include the following:

> Critic: We wish for the world to be a physical state different than it is.
> Way to Think: Seek a physical action that will make it so.
>
> Critic: There are several potential actions but it is unclear which is the best.
> Way to Think: Reject the actions that produce unacceptable consequences.
>
> Critic: We have taken an action but it has produced an unexpected outcome.
> Way to Think: Try to figure out why we failed to predict this outcome.
>
> Critic: My partner does not seem to have the same goals as I do.
> Way to Think: Try to explain how I failed to communicate my intent earlier on.

Each of these critics produces an assessment of the current overall problem situation and suggests a way to cope with it. I regard the critic-selector model as embodying a simple but very important idea: we can build AI systems not by employing rote algorithms, but by having a host of different methods, and knowledge about which method to use under which circumstances. To do this, however, we need to have some sort of language for

representing types of problems and types of solutions, as well as *types of problems with those solution methods themselves.* This thesis describes my first attempt at taking this approach to building an AI system with common sense.

1.3 A Six-Layer Model of Commonsense Thinking

The critic-selector model can be used to implement a variety of cognitive control structures, and in doing so implement different AI architectures and algorithms. In this thesis I will use it to implement a "tower of reflection" architecture for controlling the actions of actors in the artificial life domain shown in Figure 1-1. Such reflective architectures are used because it is often difficult to assure perfect operation in any one layer, and therefore a higher layer that reflects upon that layer can be added to help cope with its limitations. In this thesis I will describe a six-layer architecture with reactive, deliberative, reflective, and three types of self-reflective layers that, respectively, act, reason about that action, reflect upon that reasoning, and assess cognitive activity with respect to self-models, as shown in Figure 1-2.

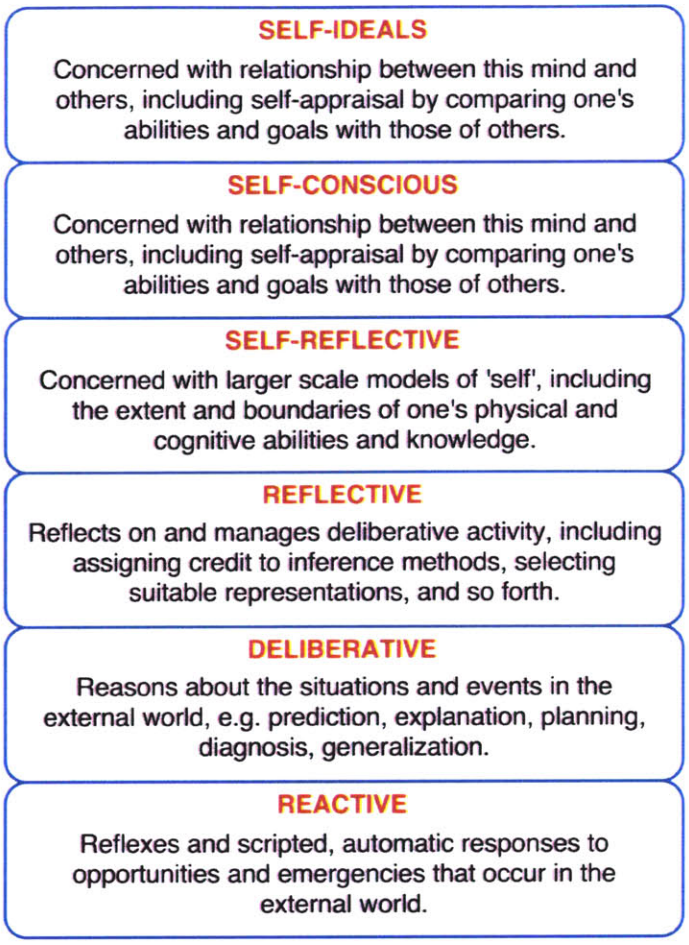


Figure 1-2. A Six-Layer Model of Commonsense Thinking

Each of these layers is populated by *mental critics* that respond to problems in the layers beneath, or in the case of the lowest reactive layer, to problems in the outside world. The reactive layer is populated by *reactive critics* that suggest courses of action based purely on the currently active goals and current observations about the state of the outside world. The deliberative layer helps the reactive layer by reasoning about the circumstances and consequences of actions proposed by reactive critics, using *deliberative critics* to produce the reasoning. The reflective layer monitors the inference processes that occur in the deliberative layer and uses *reflective critics* to assess the effectiveness of recent deliberations. The upper self-reflective, self-conscious, and self-ideals layers are populated by critics that can be used to assess actions based on criteria that has to do with

whether those actions are consistent with one's self-models. The activity of all of these critics is managed by meta-managerial critics that select subsets of the critics in these six layers by reacting to the current overall problem solving state.

1.4 Overview of EM-ONE

This six-layer model is based on Marvin Minsky's "Emotion Machine" architecture, which described in considerable detail in his book *The Emotion Machine* (Minsky, forthcoming). This thesis describes EM-ONE, an implementation of the first three layers this six-layer model (the reactive, deliberative, and reflective layers.) The major components of EM-ONE are:

1. A collection of *mental critics*:

   (a) *reactive critics* that suggest courses of action in the world.
   (b) *deliberative critics* that generate, assess, and reject hypothetical narratives.
   (c) *reflective critics* that see problems with recent actions and deliberations.

   (*Self-reflective, self-conscious, and self-ideals critics* help assess hypothetical narratives with respect to our self-models. These critics are discussed in Chapter 3 but are not part of the EM-ONE implementation.)

2. A collection of *commonsense narratives* whose contents span the physical, social, and mental realms. These narratives are used by mental critics to make analogies that help generate actions, inferences, and reflections.

3. *Meta-managerial critics* (or *metacritics* for short) that coordinate these mental critics by reacting to the current overall problem solving state.

4. A *reflective rule-based evaluator* that keeps track of all of these entities and manages the activation of meta-managerial critics and mental critics. The

evaluator makes available to critics a record of the activity of recently invoked critics, so that the system can reflect on its own performance.

EM-ONE can be thought of as a kind of "cognitive programming language" that supports the programming of reactive, deliberative, and reflective processes, and that comes with a database of commonsense knowledge in the form of commonsense narratives and a library of mental critics that apply this narrative knowledge to solve problems. Mental critics and commonsense narratives use a common narrative representation, so that critics can more easily interoperate with each other and with the commonsense knowledge base. EM-ONE also includes a sensor-effector interface to control the activity of the robots in the elemental world shown in Figure 1-1.

The following sections provide a little more detail about mental critics and commonsense narratives. These concepts and their role in the operation of EM-ONE will be discussed in great detail in Chapters 2-5 of this thesis.

1.5 Mental Critics

Mental critics are implemented as pattern matching procedures that solve problems by case-based reasoning using a library of narrative cases. Critics notice similarities between the current problem situation and narratives from the narrative case base in which similar problems occurred. Many critics are capable not only of identifying problems but also of suggesting courses of action that could help alleviate those problems. The actions critics can take are fairly open ended—they may assert new knowledge, select or suppress other critics, or even call arbitrary functions in the Common Lisp environment on top of which EM-ONE is implemented. Critics recognize problems by matching patterns encoded in a frame-based knowledge representation language that supports the description of structured scenarios involving many connected actors, actions, situations, events, objects, and properties, including mental relations such as "observes," "believes," and "desires." Critics typically take the general form shown in Figure 1-3, which is an example of a reactive critic that responds to a problem in the world by proposing a physical action to take.

```
(defcritic (reactive*difference-between-conditions-and-desires=>propose-action N)
  (in conditions current-conditions
    (observes ACTOR (not (REL SUBJ OBJ)))
    (desires ACTOR (observes ACTOR (REL SUBJ OBJ))))
  (in narratives N
    (sequential
      (observes ACTOR2 (not (REL SUBJ2 OBJ2)))
      (does ACTOR2 (ACTION ACTOR2 SUBJ2 OBJ2) [1])
      (observes ACTOR2 (REL SUBJ2 OBJ2) [2]))
    (causes [1] [2]))
  (=>)
  (in conditions current-conditions
    (assert (intends ACTOR (ACTION ACTOR SUBJ OBJ) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

Figure 1-3. Example of a Reactive Mental Critic

1.6 Commonsense Narratives

All knowledge in EM-ONE that pertains to how the physical, social, and mental world works is captured in the EM-ONE narrative corpus, a case-base of commonsense narratives. Mental critics capture generic control knowledge, but don't have very much commonsense domain knowledge in and of themselves; they rarely make references to particular types of situations, events, objects, or properties. When faced with a specific situation, they use knowledge from the EM-ONE narrative corpus to draw more specific conclusions, such as the effects of actions under different circumstances, what sorts of desires might lead an agent to take some action, and so forth. In other words, mental critics draw from the narrative corpus for ideas about how to act, deliberate, and reflect.

Representing even simple stories requires an expressive knowledge representation language. Such a language must be capable of describing a wide variety of situations, objects, actors, events, actions, properties, relations, and many other types of entities. EM-ONE narratives are represented using the same frame-based description language that mental critics use. Narratives are typically authored in a simple language-like notation that is easily parsed to produce frame-based descriptions. This notation is designed to represent stories like the following one:

Green wants to hold a stick. Green sees that Pink has a stick. Green grasps onto the stick. Pink now believes that Green desires to hold a stick. Pink releases the stick.

This story can be authored as shown in Figure 1-4.

```
(defnarrative green-grasps-stick-from-pink
  (desires green (is-holding green stick) [1])
  (sequential
    (observes green (is-holding pink stick))
    (does green (grasps green stick) [2])
    (believes pink (desires green (is-holding green stick)) [3])
    (does pink (releases pink stick) [4]))
  (causes [1] [2])
  (causes [2] [3])
  (causes [3] [4]))
```

Figure 1-4. Example of an EM-ONE Commonsense Narrative

Mental critics sometimes get at the contents of narratives indirectly, by accessing the knowledge embedded in narratives via a collection of knowledge extraction rules. For example, for the narrative in Figure 1-4, the following relations can be extracted:

- If you want to be holding an object then you should try to grasp it.
- If you grasp an object then someone may infer that you want to hold the object.
- If someone believes that you want to grasp an object they are holding, then they might choose to release the object.

These relations can then used by mental critics as sources of commonsense knowledge during the course of acting, deliberating, and reflecting.

1.7 Cognitive Activity in EM-ONE

How do all of the elements described so far work together to produce intelligent behavior? The current version of EM-ONE is built on top of CRITIC-L, a Common Lisp-based reflective rule-based evaluator for mental critics. At every "cognitive cycle," the evaluator accepts observations about the external environment and then runs all available

metacritics to think about what to do or think next. These metacritics select particular subsets of reactive, deliberative, and reflective critics in response to the present problem situation and the progress that has been made so far. One common sequencing of mental critics by metacritics is the following:

1. **Reaction.** The system first invokes reactive critics that propose possible solutions to the current problem, by observing the current situation and comparing it against narratives from the EM-ONE narrative corpus. These reactive critics propose courses of action by matching narratives in which similar goals were achieved in similar conditions by taking particular actions.

2. **Deliberation.** If actions are proposed by the reactive layer, deliberative critics are invoked to reason about the circumstances and consequences of those actions. The deliberative layer reasons by searching a space of *hypothetical narratives* starting from a seed hypothesis based on the present situation. This search is performed by iteratively applying deliberative critics that first complain about the present set of candidate hypotheses, and then proceed to spawn new hypotheses that attempt to improve upon those existing ones. The deliberative layer prefers hypotheses that are consistent with known narratives, that causally link the present situation to a clear future success or failure, that do not have major causal or explanatory gaps, that are relevant to the present context, that are rich with information, and that are internally consistent.

3. **Reflection.** The deliberative layer may make mistakes, such as producing incorrect predictions about the effects of actions. If such problems occur, then reflective critics are engaged to identify the source of these mistakes and modify the critics responsible so that they perform better in the future.

As an example of this operation, consider the interacting robots scenario of Figure 1-1. Here, reactive critics are responsible for the actual actions that are taken, e.g. moving to the table, grasping the stick, lifting the table up, etc. Deliberative critics anticipate the

consequences of the taking actions, e.g. will attempting to insert the stick under the table succeed, will grasping the stick interfere with the other actor's goals, and so forth. Reflective critics cope with mistakes in reasoning, such as mistakenly assuming that the other actor wishes to take apart the table rather than put the table together. Metacritics recognize global problem solving impasses, such as there being candidate actions whose consequences have not yet been considered, and the deliberative layer should be invoked.

Because control is managed by metacritics, this is not the only style of control that is possible. A different set of metacritics might cause the system to deliberate in advance of problems occurring (worrying), or spend no time reflecting upon recent deliberations, or turn off all the upper layers of the system so that the system becomes purely "reactive," and so forth.

1.8 Motivations: The Emotion Machine and H-CogAff

The design of EM-ONE draws heavily on Minsky's *Emotion Machine* architecture—hence the name EM-ONE—and especially the sections focusing the description of the architecture as a society of layered mental critics (Minsky, forthcoming). I have also drawn ideas from Sloman's H-CogAff architecture (Sloman, 2001), which resembles Minsky's architecture in many respects including its division into reactive, deliberative, and reflective or meta-managerial levels. Both Minsky and Sloman developed their architectures to provide rich frameworks with which to explain the diversity of complex and subtle aspects of human cognition, especially our capacity for common sense and our variety of emotions. In fact, to Minsky, common sense and emotions are both types of thinking invoked by turning on and off different collections of cognitive resources such as mental critics. Sloman has emphasized the value of the layered design to let him distinguish between such affective concepts as "emotion," "attitude," "mood," "pleasure," and so forth. My goal with EM-ONE is primarily to support more intricate forms of reflective commonsense thinking, although in the long run I hope it will help to explain a broader array of types of thinking including such feelings as love, confusion, anger, and hope.

1.9 Roadmap

In the following chapters I will elaborate on this first chapter's brief overview of EM-ONE, provide a detailed example of EM-ONE's operation, discuss related and future work, and summarize my contributions. This thesis is organized as follows:

Chapter 2: Commonsense Narratives describes the narrative-based knowledge representation used by EM-ONE.

Chapter 3: Mental Critics describes the mental critics used by EM-ONE to produce actions, deliberations, and reflections.

Chapter 4: Meta-Management describes the metacritics that coordinate the activity of mental critics to respond to different problem-types in the world and in the mind.

Chapter 5: Example Scenario demonstrates how these critics and narrative can be connected together to solve a problem in which two simulated robots cooperate to put together a table from its component parts.

Chapter 6: Related Work describes work in AI that inspired this thesis and that relates to it in other ways.

Chapter 7: Future Work describes features that I plan to add to EM-TWO, the next version of EM-ONE.

Chapter 8: Contributions summarizes my contributions and touches briefly on the future of the approach presented in this thesis.

The Appendices include details that, for the sake of continuity, I left out of the main text of the thesis.

Chapter 2

Commonsense Narratives

> *We now begin our study of the mind from within. Most books start with sensations, as the simplest mental facts, and proceed synthetically, constructing each higher stage from those below it. But this is abandoning the empirical method of investigation. No one ever had a simple sensation by itself. Consciousness, from our natal day, is of a teeming multiplicity of objects and relations, and what we call simple sensations are results of discriminative attention, pushed often to a very high degree. [...] The only thing which psychology has a right to postulate at the outset is the fact of thinking itself, and that must first be taken up and analyzed.*
>
> *– William James, Principles of Psychology (1890)*

EM-ONE represents commonsense knowledge about how the world works using *commonsense narratives*, intricate narrative structures that causally relate situations, events, objects, and their properties. EM-ONE includes a small corpus of such narratives, each describing a fragment of the physical, social, and mental activities of several actors interacting with objects and with each other in the elemental world from Figure 1-1. In this chapter I will describe how commonsense narratives are represented in EM-ONE, and provide examples of narratives that have been encoded in terms of this representation. Then, in Chapter 3, I will describe how the mental critics of EM-ONE use these narratives to propose courses of action, deliberation, and reflection in order to solve commonsense problems.

2.1 Representing Knowledge Using Narratives

Rather than representing commonsense knowledge as collections of logical rules, as is done in most present day commonsense reasoning systems (Lenat, 1995; Davis & Morgenstern, 2004), EM-ONE represents commonsense knowledge using narrative

exemplars whose contents span the physical, social, and mental realms. Here are examples, in English, of the kinds of narratives that EM-ONE uses:

- **Physical.** Pink desires that the stick is attached to the board. Pink observes that the stick is not attached to the board. Pink attaches the stick to the board. Pink now observes that the stick is attached to the board. (Captures the knowledge that the effect of attaching two objects is that they become attached.)

- **Mental.** Green desires to see the stick. Green believes that Green desires to see the stick. (Captures the knowledge that an actor often knows what it desires.)

- **Social.** Pink desires to hold the stick. Pink grasps the stick. Pink observes that Green observes that it grasped the stick. Pink believes that Green believes that Pink desires to hold the stick. (Captures the knowledge that if someone observes you take an action, then they may infer the reason you took that action.)

Compared to representing commonsense knowledge as a collection of abstract default rules, there are a number of benefits to instead representing knowledge in the form of narratives:

- **Narratives connect knowledge to purpose.** Knowledge in story form connects situations and events in the story to purposes. Rather than simply describing how, for example, an event might cause another event to occur, in a story that effect can also be connected to one of the character's goals and is seen as a source of help or hindrance. This helps in knowledge retrieval, since we can use context about present goals and circumstances to help select relevant stories.

- **Narratives help us control inference.** Knowledge in story form is organized into coherent large-scale units. One challenge with reasoning with more granular units of knowledge such as logical facts and rules is knowing what to infer—or equivalently, when to stop making inferences. Stories capture as schemas entire bounded scenarios

that can be brought to bear at once by engaging in analogical reasoning, where parts of the retrieved story are used to extend a description of the current situation, perhaps with some analogical substitutions.

- **Narratives are easy to acquire.** Knowledge in story form is easy to engineer, since one can focus on particular concrete instances rather than formulating general rules in the abstract. Someone simply has to recount details of how an episode unfolded, rather than trying to figure out precisely all the conditions *why* the episode played out that way. Despite their ease of engineering, stories are complex structures that can be used for multiple purposes and that can contain multiple types of knowledge, for example, knowledge about the effects of actions, whether states are desirable or undesirable, subgoals that help achieve supergoals, and so forth.

- **Narratives can contextualize knowledge.** Knowledge in story form is contextualized by the surrounding story. Commonsense knowledge cannot be represented as always-true rules, because there are always exceptions. Instead we must turn to rules that are only true "by default." However, using default rules raises the problems of how to decide when a default rule can be applied and when it cannot, and how to prioritize and combine different default rules. To solve these problems we must represent somewhere knowledge of the contexts in which those rules apply, which to some extent defeats the purpose of writing down a "generalized" default rule in the first place. The alternative, with stories, is to annotate the stories with the causal and logical relationships between their elements, in effect formulating default rules but embedding them within concrete contexts. Collections of stories can capture not only the more specific contexts (e.g. locations, conditions, times, etc.) in which commonsense rules apply, but also their various exceptions, and also what would otherwise be meta-information such as for what goals should those rules be applied.

Further justifications for using narrative structures have been provided by Roger Schank and his students and collaborators (e.g. Schank & Abelson (1977); Schank (1982); Schank (1986)).

One can apply the knowledge within narratives by case-based reasoning (Kolodner, 1993). In case-based reasoning, given a problem situation, one first seeks from a database of situation cases a suitably similar situation in which that problem is solved, and then adapts that solution to the present problem situation. This adaptation is often performed by generalizing the similar situation to the point where it matches the present problem situation exactly, and then replacing elements of the generalized situation with the specific details of the present problem situation, at which point the adapted solution can be applied directly to the present problem situation. In general, this process is prone to errors, because cases may be adapted based on faulty generalizations. For this reason, applying knowledge embedded within narratives can be more difficult than applying rules, because good rules are already generalized so that they apply in a wide range of specific situations.

While the solutions suggested by adapted narratives may sometimes be incorrect due to faulty adaptation, if we can find ways to pay this price, representing knowledge in narrative form can help address problems ranging from ease of acquisition, to representing context, to knowledge retrieval, to the control of inference.

2.2 The EM-ONE Narrative Representation

Representing even simple stories requires an expressive knowledge representation language. I have developed a frame-based narrative representation language for EM-ONE (Minsky, 1975). The frames of this language can be connected together in a large variety of ways to tell stories of different types. These frames support the representation of a variety of situations, events, actors, objects, properties, relations, and other entities, that can be linked together to express larger narratives. Many of the types of frames and frame slots that I used in this thesis, and their intended meanings, are listed in Table 2-1.

Table 2-1. Basic Frames Types

Frames and Frame Slots	Intended Frame Meanings
holds :sit :prop	Proposition PROP is true in situation SIT
justifies :action :hyp	ACTION is justified by hypothesis HYP
believes :actor :prop	ACTOR believes proposition PROP is true
observes :actor :prop	ACTOR observes proposition PROP to be true
expects :actor :prop	ACTOR expects proposition PROP to be true soon
observes-class :actor :class	ACTOR observes an entity of class CLASS
desires :actor :prop	ACTOR desires that proposition PROP be true
intends :actor :prop	ACTOR intends to take ACTION
does :actor :prop	ACTOR takes ACTION in the simulator
infers :actor :sit	ACTOR infers that proposition PROP is true
plans :actor :plan	ACTOR has the plan PLAN
helps :actor :other	ACTOR helps the goals of actor OTHER
hinders :actor :other	ACTOR hinders the goals of actor OTHER
engages :actor :critic	ACTOR recently engaged the criticism CRIT
moves-close-to :actor :object	ACTOR moves up close to OBJECT
grasps :actor :object	ACTOR grasps OBJECT
releases :actor :object	ACTOR releases OBJECT
lifts :actor :object	ACTOR lifts up OBJECT off the ground
waves-at :actor :other	ACTOR waves at actor OTHER
attaches :actor :object :target	ACTOR attaches OBJECT to TARGET
says :actor :expr	ACTOR says expression EXPR
fits-beneath :subject :object	SUBJECT fits underneath OBJECT
is-touching :subject :object	SUBJECT is touching OBJECT
is-reachable :subject :object	SUBJECT can reach OBJECT
is-lifting :subject :object	SUBJECT is lifting OBJECT
is-holding :subject :object	SUBJECT is holding OBJECT in its hand
is-attached :subject :object	SUBJECT is attached to OBJECT
is-above :subject :object	SUBJECT is above OBJECT
is-looking-at :subject :object	SUBJECT is looking at OBJECT
is-supporting :subject :object	SUBJECT is supporting OBJECT
is-behind :subject :object	SUBJECT is behind OBJECT
is-visible-to :subject :object	SUBJECT is visible to OBJECT

These frames span the physical, social, and mental realms: there are frames for describing physical relationships, such as concepts of nearness and support; for describing social relationships, such as concepts of helpfulness and hindrance; and for describing mental relationships, such as beliefs, desires, and intentions. As a frame-based representation, any situation, event, proposition, object, belief, or even entire narrative is a first-class object that can referenced by the slots of other frames. It is possible to include beliefs about the beliefs of other agents, and narratives that involve several degrees of nesting of constituent narratives.

These frames are represented in EM-ONE as sets of ground binary predicates that capture the set of slot relations asserted by each frame. In addition to binary predicates for the frame slots shown in Table 2-1, the EM-ONE frame language includes supplementary predicates for representing the compositional structure of situations and events, the temporal orderings that hold between them, and their causal interdependencies. These predicates are listed in Table 2-2.

Table 2-2. Supplementary Predicates

Binary Predicates	**Intended Predicate Meanings**
(subsit SIT SUBSIT)	The situation SUBSIT holds during the situation SIT
(type X TYPE)	The entity X is of type TYPE (e.g. **grasps** or **believes**)
(isa X CLASS)	The entity X is of class CLASS (e.g. **object** or **situation**)
(truth SIT TRUTH)	The situation SIT has TRUTH value (can be **true** or **false**)
(follows S1 S2)	The situation S1 is temporally followed by situation S2
(causes S1 S2)	The situation S1 causes the situation S2
(implies S1 S2)	The situation S1 implies the situation S2
(jointly S1 S2)	The situation S1 is true jointly with the situation S2
(dependency S1 S2)	The situation S2 has a dependency on the situation S1
(requirement S1 S2)	The situation S2 requires situation S1

In a logical approach to representing commonsense knowledge, these vocabulary elements would be defined by asserting logical axioms that constrained their interrelationships in particular ways. In EM-ONE, these elements are instead "defined" by example. For example, if one wishes to know what the consequences are of grasping a stick, one can look up a narrative where someone grasped a stick; the consequences might include for example that if the other agent wanted the stick, it may try to re-obtain the stick. Thus the "meanings" of these elements are determined partly by the narratives in which they play a part, and partly by the procedural effects of the mental critics (described in Chapter 3) that generate and sanction inferences using these narratives.

This frame language is not intended to be a comprehensive set of primitives for all commonsense domains. To capture all of the subtleties of the world requires far more representational apparatus than this, but because my goal is to study certain aspects of the organization of mental activity, I have compromised by using a simpler representation

scheme, so that I could make progress on these more specific issues. The vocabulary of the frame language is mainly intended to support certain types of thinking about physical, social, and mental domains useful for the table building task that I will describe in greater detail in Chapter 5.

2.3 Authoring EM-ONE Narratives

For convenience, commonsense narratives are authored in a concise, language-like notation that can easily be parsed to system of frames expressed as collections of ground binary predicates. Figure 2-1 shows an example of a narrative expressed in this concise notation:

```
(defnarrative green-grasps-stick-from-pink
  (desires green (is-holding green stick) [1])
  (sequential
    (observes green (is-holding pink stick))
    (does green (grasps green stick) [2])
    (believes pink (desires green (is-holding green stick)) [3])
    (does pink (releases pink stick) [4]))
  (causes [1] [2])
  (causes [2] [3])
  (causes [3] [4]))
```

Figure 2-1. A narrative expressed in the concise narrative notation

As in natural language, the slots to which elements are attached are implicit in their position in the statement. Unlike natural language, there is no syntactic ambiguity, so this notation can be easily parsed.[2]

The operation of the parser is straightforward. It parses each statement of the narrative in order. Each statement is parsed to a frame whose slots are filled by parsing the subexpressions of the statement. The head of the statement determines the frame type, and the remainder of the statement corresponds to the slots of the frame, where the slot

[2] In the future we may be able to encode such knowledge as English stories, and extract suitable story representations from these fragments using natural language semantic parsing methods. This would make it far easier to collect a large corpus of stories for EM-ONE to use.

type is determined by its position in the statement. The subexpressions of the statement are themselves parsed recursively to produce subframes, and these subframes are bound to the slots of the main frame. It is often the case that we would like for a frame slot to point to a previously created frame. Statements may be assigned to variables using square brackets, and these variables referenced later, as is done in the example in Figure 2-1.

After all the statements have been parsed, scaffolding frames are added to tie them together into a single narrative. Each statement's frame is embedded within a **situation** frame using a **subsit** relation. Each these situation frames are themselves embedded within the larger narrative frame also by using **subsit** relations. If frames are within a statement whose head is termed **sequential**, then between each adjacent pair of frames is asserted the **follows** temporal relationship.

As a result of this parsing process, narratives are typically hierarchically structured, where at the top-level, a narrative frame consists of a collection of constituent situation frames, which may decompose further into more granular situation frames. Situations may represent static states or dynamic events, and may involve actors and objects, possessing various properties and interrelated in various ways, and all of these entities are related by binary slot relations of various types. The final representation that is computed of the concise narrative in Figure 2-1 is the collection of ground binary predicates listed in Figure 2-2.

```
(subsit TOPSIT_19457 SIT_19459)
(truth SIT_19459 true)
(isa SIT_19459 situation)
(type SIT_19459 desires)
(actor SIT_19459 green)
(truth SIT_19460 true)
(isa SIT_19460 situation)
(type SIT_19460 is-holding)
(subsit SIT_19459 SIT_19460)
(subject SIT_19460 green)
(object SIT_19460 stick)
(prop SIT_19459 SIT_19460)
(isa SIT_19461 situation)
(subsit TOPSIT_19457 SIT_19461)
(truth SIT_19462 true)
(isa SIT_19462 situation)
(type SIT_19462 observes)
(subsit SIT_19461 SIT_19462)
(actor SIT_19462 green)
(truth SIT_19463 true)
(isa SIT_19463 situation)
(type SIT_19463 is-holding)
(subsit SIT_19462 SIT_19463)
(subject SIT_19463 pink)
(object SIT_19463 stick)
(prop SIT_19462 SIT_19463)
(truth SIT_19464 true)
(isa SIT_19464 situation)
(type SIT_19464 does)
(subsit SIT_19461 SIT_19464)
(actor SIT_19464 green)
(truth SIT_19465 true)
(isa SIT_19465 situation)
(type SIT_19465 grasps)
(subsit SIT_19464 SIT_19465)
(actor SIT_19465 green)
(object SIT_19465 stick)
(prop SIT_19464 SIT_19465)
(follows SIT_19462 SIT_19464)
(truth SIT_19466 true)
(isa SIT_19466 situation)
(type SIT_19466 believes)
(subsit SIT_19461 SIT_19466)
(actor SIT_19466 pink)
(truth SIT_19467 true)
(isa SIT_19467 situation)
(type SIT_19467 desires)
(subsit SIT_19466 SIT_19467)
(actor SIT_19467 green)
(truth SIT_19468 true)
(isa SIT_19468 situation)
(type SIT_19468 is-holding)
(subsit SIT_19467 SIT_19468)
(subject SIT_19468 green)
(object SIT_19468 stick)
(prop SIT_19467 SIT_19468)
(prop SIT_19466 SIT_19467)
(follows SIT_19464 SIT_19466)
(truth SIT_19469 true)
(isa SIT_19469 situation)
(type SIT_19469 does)
(subsit SIT_19461 SIT_19469)
(actor SIT_19469 pink)
(truth SIT_19470 true)
(isa SIT_19470 situation)
(type SIT_19470 releases)
(subsit SIT_19469 SIT_19470)
(actor SIT_19470 pink)
(object SIT_19470 stick)
(prop SIT_19469 SIT_19470)
(follows SIT_19466 SIT_19469)
(causes SIT_19462 SIT_19464)
(causes SIT_19464 SIT_19466)
(causes SIT_19466 SIT_19469)
```

Figure 2-2. The final result of parsing the EM-ONE narrative from Figure 2-1.

In this thesis, I will use the concise notation from Figure 2-1 to describe narratives, rather than the underlying binary predicate representation from Figure 2-2.

2.4 Examples of EM-ONE Narratives

Some examples of EM-ONE narratives about the robots Green and Pink from the introduction are given below.

1. Pink desires that the stick be attached to board. Pink observes that the stick is not attached to the board. Pink attaches the stick the board. Pink observes that the stick is attached to the board. This latter observation was caused by Pink's act of attaching the stick to the board.

```
(defnarrative attaching-stick
  (desires pink (is-attached stick board))
  (sequential
    (observes pink (not (is-attached stick board)))
    (does pink (attaches pink stick board) [1])
    (observes pink (is-attached stick board) [2]))
  (causes [1] [2]))
```

2. Pink observes a table, but Pink also sees four sticks attached to a board, and these two ways to look at the situation imply each other.

```
(defnarrative table-made-of-components
  (together
    (observes-class pink table [1])
    (observes pink (is-attached stick1 board) [2])
    (observes pink (is-attached stick2 board) [3])
    (observes pink (is-attached stick3 board) [4])
    (observes pink (is-attached stick4 board) [5]))
  (jointly [1] [2])
  (jointly [1] [3])
  (jointly [1] [4])
  (jointly [1] [5]))
```

3. Pink desires that the stick is visible. This implies that Pink believes that Pink desires that stick is visible.

```
(defnarrative knowing-of-belief
  (together
    (desires pink (is-visible-to pink stick) [1])
    (believes pink [1] [2]))
  (implies [1] [2]))
```

4. Pink wants to be holding the stick. Pink moves close to the stick. Pink then observes that Green grasps the stick. Pink does not want Green to be holding the stick, and so Pink complains by saying "No."

```
(defnarrative green-interferes-with-pink
  (desires pink (is-holding pink stick) [1])
  (sequential
    (does pink (moves-close-to pink stick) [2])
    (observes pink (grasps green stick))
    (desires pink (not (is-holding green stick)) [3])
    (does pink (says pink No) [4]))
  (causes [1] [2])
  (causes [3] [4]))
```

5. Pink desires to observe a table, but Pink does not observe one. Pink thus plans to assemble a table. Pink believes that assembling a table requires a stick. But Pink does not observe a stick, and as a result Pink abandons its desire to observe a table.

```
(defnarrative pink-abandons-goal-because-missing-requirements
  (sequential
    (desires pink (observes-class pink table) [1])
    (not (observes-class pink table [2]))
    (plans pink assembles-table [3])
    (believes pink (requires assembles-table stick))
    (believes pink (not (is-visible-to pink stick)) [4])
    (desires pink (not (observes-class pink table)) [5]))
  (dependency [1] [3])
  (dependency [2] [3])
  (dependency [4] [5]))
```

6. Pink believes that Green desires to be holding a stick. Pink observes that Green releases the stick it was holding and moves away from the stick. Pink now believes that Green does not desire to be holding a stick.

```
(defnarrative pink-revises-its-belief
  (sequential
    (believes pink (desires green (is-holding green stick)))
    (observes pink (releases green stick) [1])
    (observes pink (moves-away-from green stick) [2])
    (believes pink (not (desires green (is-holding green stick))) [3]))
  (dependency [1] [3])
  (dependency [2] [3]))
```

2.5 Commonsense Knowledge Embedded Within Narratives

Mental critics sometimes use the content of EM-ONE narratives indirectly, using a collection of auxiliary generalization rules to extract particular facts or dependencies that

exist within narratives. There are many types of commonsense knowledge embedded within these narratives. For example, consider the following story:

```
(defnarrative green-grasps-stick-from-pink
  (desires green (is-holding green stick) [1])
  (sequential
    (observes green (is-holding pink stick))
    (does green (grasps green stick) [2])
    (believes pink (desires green (is-holding green stick)) [3])
    (does pink (releases pink stick) [4]))
  (causes [1] [2])
  (causes [2] [3])
  (causes [3] [4]))
```

Embedded in the story are the following default rules about the world:

- If you want to be holding an object then you should try to grasp it.
- If you grasp an object then someone may infer that you want to hold the object.
- If someone believes that you want to grasp an object they are holding, then they might choose to release the object.

This type of generalizing from a single example is not always reliable. Just because there exists a pattern in a single example does not mean there exists in general the dependency suggested by the pattern, because that example might represent the exception rather than the rule. However, in EM-ONE the process of generalization is simplified because the dependencies between story elements are in many cases explicitly represented.[3] In this regard, EM-ONE narratives resemble a collection of logical constraints embedded within a concrete context.[4]

---

[3] However, even if these dependencies were not provided explicitly, given a "good story"—one where all the events have a reasonable chance of being causally related—I suspect it is often possible to extract reasonable generalizations.

[4] In Cyc (Lenat, 1995), all knowledge is kept within separate databases called contexts (which are internally consistent but mutually potentially inconsistent), but the knowledge in a Cyc context is generally a collection of abstract default rules, as opposed to a description of a concrete story annotated with dependencies relating the specific elements of the story. It is easy to represent EM-ONE narratives within Cyc, but this is not a conventional way to represent knowledge within Cyc.

EM-ONE includes a number of auxiliary *extraction predicates* defined to simplify extracting useful fragments of knowledge from narratives by making it easier to encode the antecedent portion of mental critics. Here are some of the extraction predicates used by the critics of EM-ONE:

Relations-Hold-Together. Certain relations between objects are often observed to hold simultaneously.

```
(defextractor (relations-hold-together REL1 REL2)
  (together
    (observes ACTOR (REL1 SUBJ OBJ) [1])
    (observes ACTOR (REL2 SUBJ OBJ) [2]))
  (jointly [1] [2]))
```

Relation-Causes-Relation. A given relation holding between two objects often causes another relation to hold in the following situation.

```
(defextractor (relation-causes-relation REL1 REL2)
  (sequential
    (observes ACTOR (REL1 SUBJ OBJ) [1])
    (observes ACTOR (REL2 SUBJ OBJ) [2]))
  (causes [1] [2]))
```

Effect-Of-Action-On-Object-Property. Taking an action on an object causes a property of the object to change to a different value.

```
(defextractor (effect-of-action-on-object-property ACTION PROP OBJ VALUE1 VALUE2)
  (sequential
    (together
      (observes ACTOR (PROP OBJ VALUE1))
      (does ACTOR (ACTION ACTOR OBJ))
      [1])
    (observes ACTOR (PROP OBJ VALUE2) [2]))
  (prolog (not (= VALUE1 VALUE2)))
  (causes [1] [2]))
```

Action-Achieves-Relation. Taking an action involving two objects causes a relation between those objects to come to hold true.

```
(defextractor (action-achieves-relation ACTION REL SUBJ OBJ)
  (sequential
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [ACTION])
    (observes ACTOR (REL SUBJ OBJ) [RESULT]))
  (causes [ACTION] [RESULT]))
```

Effect-of-Action-Is-New-Object. Taking an action results in the creation of a new object of a given type.

```
(defextractor (effect-of-action-is-new-object ACTION OBJECT_TYPE)
  (sequential
    (together
      (not (observes-class ACTOR OBJECT_TYPE))
      (does ACTOR (ACTION ACTOR))
      [1])
    (observes-class ACTOR OBJECT_TYPE [2]))
  (causes [1] [2]))
```

Action-Requires-Precondition. For an action to successfully produce a certain effect relation between two objects, a given relation must already hold between those objects.

```
(defextractor (action-requires-precondition ACTION PRECOND SUBJ OBJ)
  (sequential
    (together
      (observes ACTOR (PRECOND SUBJ OBJ))
      (does ACTOR (ACTION ACTOR SUBJ OBJ))
      [1])
    (observes ACTOR (EFFECT SUBJ OBJ) [2]))
  (requirement [1] [2]))
```

Actor-Desires-Situation. An actor may desire a situation in which a particular relation holds between two objects.

```
(defextractor (actor-desires-situation ACTOR REL SUBJ OBJ)
  (desires ACTOR (observes ACTOR (REL SUBJ OBJ))))
```

2.6 Summary

This chapter described (a) the narrative knowledge representation scheme that EM-ONE uses to represent commonsense knowledge, (b) how narratives can be authored using the **defnarrative** operator, and (c) how the **defextractor** operator can be used to define

extraction predicates that make it more convenient to draw useful fragments of knowledge from instances of these narratives. In the next chapter I will describe how EM-ONE engages mental critics to use these narratives to suggest possible courses of action, deliberation, and reflection to solve commonsense problems.

Chapter 3

Mental Critics

> *The claim that a machine cannot be the subject of its own thought can of course only be answered if it can be shown that the machine has some thought with some subject matter. Nevertheless, "the subject matter of a machine's operations" does seem to mean something, at least to the people who deal with it. If, for instance, the machine was trying to find a solution of the equation x2 - 40x - 11 = 0 one would be tempted to describe this equation as part of the machine's subject matter at that moment. In this sort of sense a machine undoubtedly can be its own subject matter. It may be used to help in making up its own programmes, or to predict the effect of alterations in its own structure. By observing the results of its own behaviour it can modify its own programmes so as to achieve some purpose more effectively. These are possibilities of the near future, rather than Utopian dreams.*
>
> *– Alan Turing, Computing Machinery and Intelligence (1950)*

Cognitive activity in EM-ONE is produced by *mental critics*. Mental critics are procedures that recognize types of problems in the world and in the mind of EM-ONE itself, and act to resolve those problems. Mental critics operate to produce actions in the world, make inferences by constructing hypothetical scenarios elaborating on what is known, and reflect on the system's own recent thinking to try to improve future thinking. The application of mental critics is itself directed by a collection of top-level "manager" critics called *metacritics*, which will be described more fully in Chapter 4. This chapter describes six types of mental critics: reactive, deliberative, reflective, self-reflective, self-conscious, and self-ideals mental critics.

3.1 What is a Mental Critic?

The basic idea behind mental critics is essentially that behind all error-driven adaptive systems. Critics recognize errors of various types and activate ways to try to eliminate or lessen them. I have taken this simple idea and applied it as a general principle for control throughout the EM-ONE system. While this basic idea is very old within AI, going back to the General Problem Solver (Newell, Shaw, & Simon, 1960a), it has largely been used as a way to write programs that solve problems in the outside world, as opposed to solve problems *within the system itself.* In EM-ONE, many mental critics react to problems with the activity of other mental critics.

In the following sections I will describe each these types of critics in more detail:

Reactive Critics. The critics in the reactive layer interface directly with the external environment. For example, we might have a reactive critic that notices that a stick is on the ground and that we wish to be holding a stick, and therefore proposes that we try to pick up the stick. This reactive critic proposes a course of action by recognizing a difference between the currently observed situation and the currently active goals, and recognizing that in one of the narratives in the narrative corpus this difference was reduced by applying a particular method. No further deliberation is involved, and in fact, it is possible to encode a wide variety of behaviors purely by applying reactive critics alone.

Deliberative Critics. The critics in the deliberative layer operate on representations of the world and of the actions proposed by the reactive layer. Deliberative critics operate on narrative hypotheses, which are identical to the narratives in the EM-ONE narrative corpus except that they are developed and elaborated during the course of deliberation. Deliberative critics can be used to assess hypotheses according to criteria such as whether the actors in them are taking actions that make sense with respect to their goals, or whether the hypotheses are consistent with known commonsense narratives. Deliberative critics can ameliorate these critical assessments by producing new hypotheses that are improvements over existing ones.

These improvements are often made by drawing analogies to narratives from the EM-ONE narrative corpus.

Reflective Critics. The critics in the reflective layer operate on traces of recent deliberation and action, in other words, representations of the activity within the mind of the system itself. Reflective critics recognize problems in the system's recent activities including having made mistaken assumptions and jumping to inappropriate conclusions. These critics are capable of modifying the critics responsible for making these mistakes so that these mistakes are not repeated.

Self-Reflective, Self-Conscious, and Self-Ideals Critics. The critics in these "upper reflective" layers, not yet implemented in EM-ONE, are intended to be used primarily by the deliberative layer when it produces assessments of narrative hypotheses that involve descriptions of the system itself as an actor in the narrative. Self-reflective critics recognize limits in the abilities of the system itself, and will complain if a hypothesis involves the system taking some action it is not capable of. Self-conscious critics recognize problems that have to do with the relationship between the system's self-models and its estimation of how others view it, for example, if the system fails at an action that the other actor assumes it could easily succeed at. Self-ideals critics recognize problems where the system's actions and thinking are inconsistent with its higher-level values and ideals, for example, a "golden rule" self-ideals critic would complain if the system acts towards another actor in a way that it would not have wanted the other actor to act towards the system.

The following type of critic will be discussed in Chapter 4.

Meta-Managerial Critics. The critics in the meta-managerial layer have global access to all the other critics in the system and their activities. Metacritics select collections of critics, possibly encompassing entire layers, based on assessments of the present situation and the progress the system has made so far. For example, a metacritic

might select the reactive layer if no action has been proposed and there are many pressing goals.

In the following sections I will describe in more detail the types and structures of reactive, deliberative, and reflective mental critics, and say a little more about self-reflective, self-conscious, and self-ideals mental critics.

3.2 The CRITIC-L System

The selection, evaluation, and application of critics is managed by CRITIC-L, a collection of macros, functions, and data structures built on top of the standard Common Lisp evaluator. CRITIC-L is essentially a rule-based system that keeps track of the system's current situation and desires, the set of narrative hypotheses that the deliberative layer is presently considering, the record of recent observations made by the reactive layer and actions that were taken in the world, and the history of all activations of mental critics, including any modifications they might have made to mental state. In addition, CRITIC-L manages the storage and retrieval of the narratives of the EM-ONE narrative corpus.

Critics are implemented as large pattern matching rules based on the same representation as the EM-ONE narratives. Critics are typically implemented as case-based reasoning rules, in that while some of their antecedent elements match against the present conditions, other elements match against narratives in the narrative corpus, and the solutions that critics propose may involve elements drawn from both of these sources.

Figure 3-1 shows an example of how one type of reactive critic is authored in CRITIC-L. Note that uppercase symbols are variable names. (While I do always include the colon-prefixed slot names when encoding critics, I will ignore them from this point on in this document to avoid cluttering up the text. As with narratives, the presence of slots is not difficult to infer.)

```
(defcritic (reactive*difference-between-conditions-and-desires=>propose-action N)
  (in conditions current-conditions
    (observes :actor ACTOR :prop (not (REL :subject SUBJ :object OBJ)))
    (desires :actor ACTOR
             :prop (observes :actor ACTOR
                             :prop (REL :subject SUBJ :object OBJ))))
  (in narratives N
    (sequential
      (observes :actor ACTOR2 :prop (not (REL :subject SUBJ2 :object OBJ2)))
      (does :actor ACTOR2
            :prop (ACTION :actor ACTOR2 :object SUBJ2 :target OBJ2) [1])
      (observes :actor ACTOR2 :prop (REL :subject SUBJ2 :object OBJ2) [2]))
    (causes [1] [2]))
  (=>)
  (in conditions current-conditions
    (assert
      (intends :actor ACTOR
               :prop (ACTION :actor ACTOR :object SUBJ :target OBJ) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

Figure 3-1. An Example of a Reactive Mental Critic

If all of the conditions on the antecedent side match, the action on the consequent side is taken. This action is typically a set of knowledge base assertions or retractions, or lisp operations. Their effects may include taking an action in the world, formulating a new hypothesis that elaborates an existing one, modifying the structure of a critic, or calling other critics. Some critics do not have a (=>) symbol, in which case the critic is treated as having only a consequent side that is run like an ordinary Common Lisp function. Some critics end with a (=>) symbol, but have an empty consequent side; such critics are generally used only for assessing the quality of hypotheses producing during deliberation.

Critics interact with four separate databases of facts, each of which has a different role to play in the EM-ONE architecture:

- **Narratives** is the database of the EM-ONE narrative corpus. For the moment, most critics draw from only a single narrative at a time, but it is possible to define a critic that draws elements from two or more EM-ONE narratives.

- **Conditions** is the database of facts that represents the presently perceived sensory data, as well as the current beliefs, desires, and intentions of the actors.

- **Hypotheses** is the database of narrative hypotheses that is used by the deliberative layer while making inferences.

- **Reflections** is the database that stores the trace of invocations of critics and their effects.

Each of these databases is further divided into a collection of *contexts*, which are each a set of facts treated as a unit. For example, each narrative in the EM-ONE narrative corpus is a represented as a context within the **narratives** database. As shown in the critic from Figure 3-2, the **in** operator selects which database and context a given pattern should match against or be asserted into.

3.2.1 Running mental critics

Mental critics can be run in two ways: evaluation and application. *Evaluating* a critic matches all its antecedent elements but does not run the procedures specified on its consequent side. This is useful if we wish to assess a narrative hypothesis, for example, by counting how many deliberative critics match that hypothesis. *Applying* a critic not only matches all its antecedent elements, but also runs the procedures specified on its consequent side. Ideally, running those procedures will lead to this critic no longer matching because its effects successfully deal with the criticism.

3.2.2 Naming mental critics

I employ a naming convention for critics that I have found makes understanding their operation a little easier. Critic names provide information about what layer of the architecture they are in, what problem symptom they detect, what bug they attribute the problem symptom to, and what method they use to address the problem. The convention for commonly used critic types is listed in Table 3-1.

Table 3-1. Critic Name Patterns

Critic Name Pattern	Example Critic Name
layer*symptom*bug=>repair	reflective*expectation-failure*false-assumption=>modify-critic
layer*symptom=>repair	deliberative*unknown-motivation=>infer-motivation-by-analogy
layer=>repair	meta=>engage-reactive-layer
layer*symptom*bug	reflective*expectation-failure*assumed-wrong-actor-intention
layer*symptom	deliberative*inconsistent-actor-beliefs

Again, not all critics actually suggest a way to solve the problems they detect. As will be further discussed in the section on deliberative critics, this is because many critics are used only to assess hypotheses that are generated during deliberation, and their role is only to produce criticisms that are then used to decide which hypotheses should be accepted and which rejected.

3.2.3 The CRITIC-L implementation

Presently CRITIC-L is a sublanguage within Common Lisp. The **defcritic** operator is implemented as a Common Lisp macro that expands the given critic expression to a more complicated Common Lisp expression that produces several compiled Common Lisp functions. (More details about the form of this expansion are given in Appendix A.) Critics can be called just like ordinary Common Lisp functions. The compiled produced by **defcritic** make extensive use of Allegro Prolog, a version of Prolog embedded within Common Lisp.[5] I should stress that I do not use the underlying Prolog process for sophisticated forms of commonsense reasoning—it is only there to provide a backward chaining pattern matching functionality, so that critics can include recursively matched auxiliary predicates (including the auxiliary extraction predicates described in Chapter 2) as pattern elements. It would not be difficult to implement EM-ONE on top of other rule-based substrates.[6]

[5] Allegro Prolog is part of the commercial Allegro Common Lisp environment sold by Franz Inc.

[6] In a future version of EM-ONE I plan to move to softer forms of matching that allow M-of-N or statistically most probable matches.

3.3 Reactive Critics

Reactive critics respond to problems that can be directly sensed in the world by proposing actions to take in the world. Typically, reactive critics respond to the current conditions, which include observations about the present state of the world, the presently active collection of desires, and beliefs that may have been derived from earlier inference. Some reactive critics propose actions to take directly, but many propose actions by analogy to actions taken within narratives from the EM-ONE narrative corpus. Reactive critics by themselves are enough to produce some minimal level of competent behavior by the architecture.

3.3.1 Examples of Reactive Critics

The following are examples of reactive critics used by EM-ONE.

Reactive*Difference-Between-Conditions-and-Desires=>Propose-Action-by-Analogy. There is a difference between the observed situation and a desired goal. There exists a narrative in which that difference was reduced by taking an action. Propose to take that action.

```
(defcritic (reactive*difference-between-conditions-and-desires
                    =>propose-action-by-analogy N)
  (in conditions current-conditions
    (observes ACTOR (not (REL SUBJ OBJ)))
    (desires ACTOR (observes ACTOR (REL SUBJ OBJ))))
  (in narratives N
    (sequential
      (observes ACTOR2 (not (REL SUBJ2 OBJ2)))
      (does ACTOR2 (ACTION ACTOR2 SUBJ2 OBJ2) [1])
      (observes ACTOR2 (REL SUBJ2 OBJ2) [2]))
    (causes [1] [2]))
  (=>)
  (in conditions current-conditions
    (assert (intends ACTOR (ACTION ACTOR SUBJ OBJ) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

Reactive*Special-Observation=>Act-Reflexively. The other actor has called for help. Turn towards the actor immediately.

```
(defcritic (reactive*special-observation=>act-reflexively)
  (in conditions current-conditions
    (observes ACTOR (says OTHER "Help")))
  (prolog (not (= ACTOR OTHER)))
  (=>)
  (in conditions current-conditions
    (assert (does ACTOR (turns-toward ACTOR OTHER) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

Reactive=>Explicitly-Communicate-Intent. The actor is pursuing a plan. It believes that another actor is watching it. The actor demonstrates to the other that it is pursuing this plan by taking one action from the plan.

```
(defcritic (reactive=>explicitly-communicate-intent)
  (in conditions current-conditions
    (plans ACTOR PLAN)
    (observes ACTOR (is-visible-to OTHER ACTOR)))
  (prolog (not (= ACTOR OTHER)))
  (action-in-plan PLAN ACTION SUBJ OBJ)
  (=>)
  (in conditions current-conditions
    (assert (does ACTOR (ACTION ACTOR SUBJ OBJ) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

Reactive=>Take-Action. Causes an intended action to actually be taken in the world.

```
(defcritic (reactive=>take-action)
  (in conditions current-conditions
    (intends ACTOR (ACTION ACTOR SUBJ OBJ)))
  (=>)
  (in conditions current-conditions
    (assert (does ACTOR (ACTION ACTOR SUBJ OBJ) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

3.4 Deliberative Critics

It is well known that reactive processes, while useful for providing real-world responsiveness for the most common and familiar situations, often run into trouble when faced with novel situations. This is because no reasonably sized fixed table of responses can anticipate and account for all the potential contexts in which an action might be taken. The deliberative layer of EM-ONE helps the reactive layer produce successful actions by engaging in additional reasoning about the action, and the present

circumstances in general, prior to its being taken. This reasoning is performed by deliberative critics.

3.4.1 How deliberation helps reaction

While reactive critics by themselves are enough to produce some minimal level of competent behavior by the architecture, they may encounter several kinds of problems:

- They may not be able to retrieve a solution to the given problem.
- They may retrieve multiple solutions and not know how to choose between them.
- They may propose an action, but the action fails because adequate conditions for the action's success do not hold. Determining whether adequate conditions hold is sometimes called the *qualification problem.*
- They may propose an action, and that action succeeds at achieving its immediate primary effect, but additional undesirable consequences ensue as well. Determining the important effects of actions is sometimes called the *ramification problem.*

The deliberative layer can help to ameliorate these problems by reasoning about the actions proposed by the reactive layer. This may produce inferences that eventually lead to these actions being taken, adapted somehow, or suppressed. For example, the deliberative layer might help verify that the conditions of the reactive action indeed lead to the desired effects. If they do not, then perhaps the conditions can be changed (in effect subgoaling), or perhaps the parameters of the action can be modified. Or, they might help verify that the reactive action does not have additional unintended side effects, bringing to bear knowledge about what states of the world are undesirable and what the effects of actions typically are in the current context. If the action does cause trouble, then perhaps it should be suppressed, or perhaps it should be modified by making an analogy to a more successful attempt. In addition to being concerned about our own actions, we may wish to reason about the actions of other actors, and wonder if the actions taken by the other actors might cause trouble for us. These are only a few examples of what the deliberative layer might do; there are an enormous number of possible types of deliberations.

In EM-ONE, the deliberation process is cast as a controlled search through the space of hypothetical narratives. These hypotheses are structured just like the EM-ONE narratives, except that they are generated dynamically during the course of deliberation. Unlike reactive critics, deliberative critics operate not on the presently observed sensory state, but instead upon hypotheses. These hypotheses are produced initially by metacritics that seed the deliberative layer with an initial hypothesis based on present observations, and later produced by other deliberative critics during the deliberation process. The overall search process is guided to prefer the production of *plausible, important, cohesive, relevant, informative,* and *consistent* hypotheses, as will be discussed below.

3.4.2 The structure and function of deliberative critics

Deliberation is performed by the coordinated operation of deliberative critics. Deliberative critics identify problems with hypotheses, and suggest modifications to those hypotheses that ameliorate those problems. Consider the example in Figure 3-2.

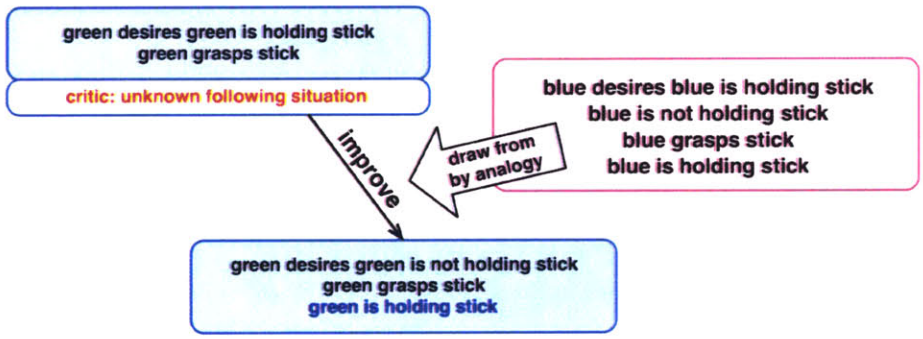


Figure 3-2. How a deliberative critic can improve a hypothesis by analogy.

Here, a deliberative critic complains that an action is mentioned in the given hypothetical narrative, but there is no description of its consequences. It then constructs a new hypothetical narrative by analogy to one of the stories in the EM-ONE narrative corpus in which that action does have a known consequence. This particular deliberative critic is authored as follows:

```
(defcritic (deliberative*unknown-action-consequence*hypothesize-by-analogy H N)
  (in hypotheses H
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [BEFORE])
    (not (follows [BEFORE] SIT)))
  (in narratives N
    (sequential
      (does ACTOR2 (ACTION ACTOR2 SUBJ2 OBJ2) [ACTION])
      (observes ACTOR2 (REL2 SUBJ2 OBJ2) [RESULT]))
    (causes [ACTION] [RESULT]))
  (=>)
  (lisp NEW_H (extend-hypothesis H))
  (in hypotheses NEW_H
    (assert (observes ACTOR (REL2 SUBJ OBJ) [[S]]))
    (assert (follows [BEFORE] [[S]]))
    (assert (subsit NEW_H [[S]]))))
```

(Note that on the consequent side, variable names that refer to newly asserted frames are created using double square brackets.)

Further examples of deliberative critics will be provided in section 3.4.5, but first let us discuss how the overall process of deliberation works in EM-ONE.

3.4.3 The cyclical process of deliberation

Deliberation proceeds via a cyclical process that operates in three phases: (a) hypotheses are assessed by evaluating deliberative critics, (b) the hypotheses with the least potential are filtered out, and (c) improved variations of the remaining hypotheses are generated by applying deliberative critics.[7]

Phase 1: Assessing hypotheses

Hypotheses are assessed according to the many criteria that are computed by deliberative assessment critics. These assessments can be divided roughly into the following general categories:

---

[7] In my thesis proposal, I discuss the idea that deliberative thinking is a kind of mental "brainstorming", where deliberation proceeds via a dialectic process of iterating between positive, generative processes, and negative, critical processes. In EM-ONE, critics play both roles—first they recognize a problem, and then they suggest a potential solution. I think of each critic as a unit of "micro-brainstorming."

Plausibility. The first form of assessment is *plausibility assessment*, where hypotheses are assessed based on whether they resemble and are consistent with known narratives.

Importance. The second form of assessment is *importance assessment*, where hypotheses are assessed based on whether or not the actors in them take actions that lead to a definite success or failure. Hypotheses that are highly desirable or highly undesirable are kept around, the former as potential solutions and the latter as scenarios to avoid. Hypotheses where the actions have no important consequences are considered "uninteresting" and filtered out.

Cohesiveness. The third form of assessment is *cohesiveness assessment*, where hypotheses are assessed based on whether they involve groups of causally connected elements. Deliberation in EM-ONE aims to produce hypotheses where elements of hypotheses are to some extent justified by the other elements.

Relevance. The fourth form of assessment is *relevance assessment*, where hypotheses are assessed based on whether they apply to the present situation. In EM-ONE, relevance is easy to obtain because the deliberative layer is generally seeded with an initial hypothesis based on a description of the current situation, and new hypotheses are largely elaborations of this initial seed.

Informativeness. The fifth form of assessment is *informativeness assessment*, where hypotheses are assessed based on whether they are missing useful information. Many deliberative critics respond to perceived gaps in a hypothesis, for example, by inferring unstated consequences of actions or unstated motivations of actors.

Consistency. The sixth form of assessment is *consistency assessment*, where hypotheses are assessed based on whether they are internally consistent. This form of assessment rejects hypotheses in which, for example, there exist situations in which relations are observed both to hold and not to hold.

To summarize this, the hypotheses that are preferred are the ones that are consistent with known narratives, that causally link situations to clear future success or failure, that do not have major causal or explanatory gaps, that are relevant to the present context, that are rich with information, and that are internally consistent.

Phase 2: Filtering poor hypotheses

Deliberative critics can spawn enormous numbers of potential elaborated hypotheses—we cannot consider every possible hypothesis that is generated. After hypotheses are assessed, they are filtered, producing a kind of best first search through the space of hypothetical narratives. This requires some strategy for using the produced assessments to compare and filter the produced hypotheses. In principle this is itself a process that could engage further deliberation. However, EM-ONE uses the simpler method of combining the produced assessments into a score that can be used to rank the present set of hypotheses. In the long run, the scoring mechanism should take into account that different criticisms have varying levels of urgency and hardness to overcome them.

Phase 3: Generating improved hypotheses

Applying a deliberative critic to a given hypothesis and narrative causes it to use the narrative to generate a new hypothesis, often by adding new elements from the narrative into the given hypothesis by analogical substitutions. These analogies may result in different types of elaborations to the hypothesis, producing an explanation for an action, a prediction of a subsequent event, a classification of a situation, and so forth. These new hypotheses do not have to be completely sound because later critics may reject generated hypotheses that do not make sense to them.

I sometimes refer to this cyclical process of deliberation as "brainstorming," as it separates out the critical assessment of hypotheses from the development of new hypotheses to resolve those criticisms.

3.4.4 Overall process of deliberation

Figure 3-3 depicts the overall process of deliberation in EM-ONE.

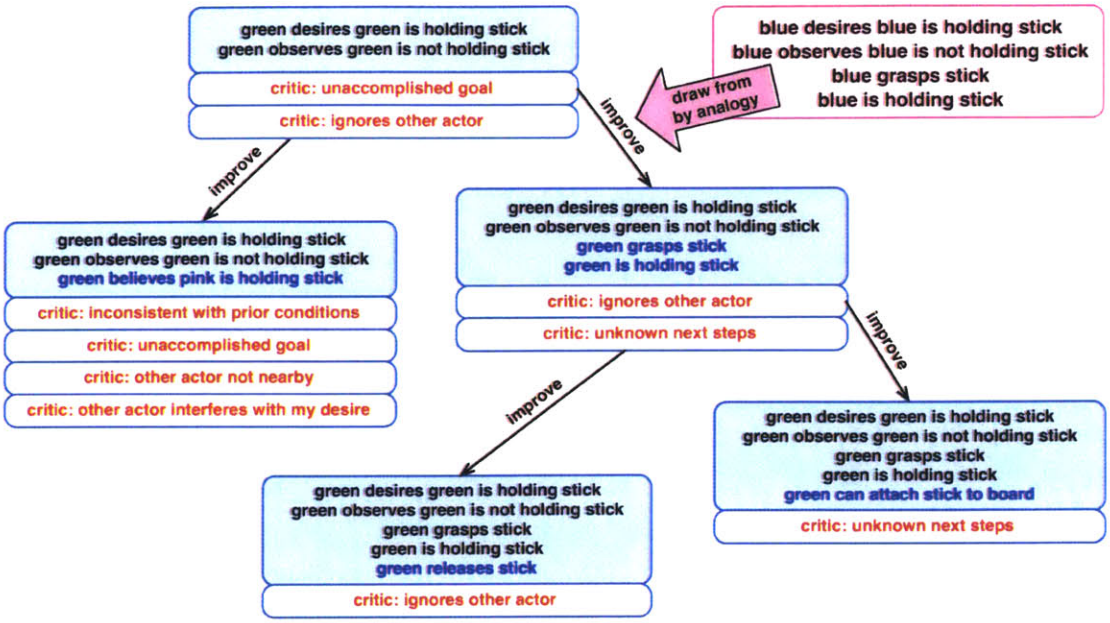


Figure 3-3. Deliberation in EM-ONE. Metacritics plant seed hypotheses in the deliberative layer, which are then criticized and improved by deliberative critics.

Because the search space of possible narrative hypotheses is enormous, the selection of deliberative critics is guided procedurally by metacritics, an approach that will be discussed in Chapter 4. This method of deliberation is neither sound nor complete, which leads to errors such as incorrect inferences being made and correct inferences failing to be made. Section 3.5 discusses how the use of reflection can help ameliorate some such errors.

One way to think about the operation of the deliberative layer is as seeking out proofs that provide answers to questions asked by deliberative critics that complain about missing information, such as the effect of a proposed action, or the motivation of the other actor. These answers should be inferable from observations and from background commonsense knowledge via trustworthy chains of inference (this is the reason for cohesiveness assessment.) By applying deliberative critics, the deliberative layer seeks out justified answers to these questions. The space of narrative hypotheses has some

elements of both model space (descriptions of concrete situations) and proof space (descriptions of the dependency relationships between these situations.)

3.4.5 Examples of deliberative critics

The following are examples of deliberative critics used by EM-ONE.

Deliberative*Unknown-Action-Consequence=>Hypothesize-By-Analogy. We have observed a situation where an event occurs. There is no description of what might follow this event. Hypothesize what might occur following this event by analogy.

```
(defcritic (deliberative*unknown-action-consequence*hypothesize-by-analogy H N)
  (in hypotheses H
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [BEFORE])
    (not (follows [BEFORE] SIT)))
  (in narratives N
    (sequential
      (does ACTOR2 (ACTION ACTOR2 SUBJ2 OBJ2) [ACTION])
      (observes ACTOR2 (REL2 SUBJ2 OBJ2) [RESULT]))
    (causes [ACTION] [RESULT]))
  (=>)
  (lisp NEW_H (extend-hypothesis H))
  (in hypotheses NEW_H
    (assert (observes ACTOR (REL2 SUBJ OBJ) [[S]]))
    (assert (follows [BEFORE] [[S]]))
    (assert (causes [BEFORE] [[S]]))
    (assert (subsit NEW_H [[S]]))))
```

Deliberative*Unknown-Relation-Consequence=>Hypothesize-By-Analogy. We have observed a situation where a relation holds. There is no description of what might follow this situation. Hypothesize what could occur following this event by analogy.

```
(defcritic (deliberative*unknown-relation-consequence*hypothesize-by-analogy H N)
  (in hypotheses H
    (observes ACTOR (REL1 SUBJ OBJ) [BEFORE])
    (not (follows [BEFORE] SIT)))
  (in narratives N
    (sequential
      (observes ACTOR2 (REL1 SUBJ2 OBJ2) [R1])
      (observes ACTOR2 (REL2 SUBJ2 OBJ2) [R2]))
    (causes [R1] [R2]))
  (=>)
  (lisp NEW_H (extend-hypothesis H))
  (in hypotheses NEW_H
    (assert (observes ACTOR (REL2 SUBJ OBJ) [[S]]))
    (assert (follows [BEFORE] [[S]]))
    (assert (causes [BEFORE] [[S]]))
    (assert (subsit NEW_H [[S]]))))
```

Deliberative*Unknown-Motivation=>Hypothesize-By-Analogy. We have observed a situation where an actor takes an action. There is no description of what might have motivated the actor to take that action. Hypothesize the actor's motivation by analogy.

```
(defcritic (deliberative*unknown-motivation=>hypothesize-by-analogy H N)
  (in hypotheses H
    (observes ACTOR (does OTHER (ACTION OTHER SUBJ OBJ)) [ACT])
    (not (desires OTHER PROP)))
  (action-achieves-relation ACTION REL SUBJ OBJ N)
  (=>)
  (lisp NEW_H (extend-hypothesis H))
  (in hypotheses NEW_H
    (assert (desires OTHER (REL SUBJ OBJ) [[S]]))
    (assert (causes [[S]] [ACT]))
    (assert (subsit NEW_H [[S]]))))
```

Deliberative*Sequential-Observation-Inconsistent-With-Dependency. In a given hypothesis, a sequence of observations is not consistent with a dependency expressed in a given narrative. This sort of critic can be used to assess the plausibility of a given hypothesis with respect to existing commonsense knowledge about the causal relationships between events.

```
(defcritic (deliberative*sequential-observation-inconsistent-with-dependency H N)
  (in narratives N
    (sequential
      (observes ACTOR (REL1 SUBJ1 OBJ1 TRUTH1) [S1])
      (observes ACTOR (REL2 SUBJ2 OBJ2 TRUTH2) [S2]))
    (causes S1 S2))
  (lisp OPPOSITE (if (eq TRUTH2 'true) 'false 'true))
  (in hypotheses H
    (sequential
      (observes ACTOR (REL1 SUBJ1 OBJ1 TRUTH1) [S1])
      (observes ACTOR (REL2 SUBJ2 OBJ2 OPPOSITE) [S2])))
  (=>))
```

Deliberative*Observes-Opposite-Of-Actor-Desire. In a given hypothesis, it is observed that one of the actors present desires is not achieved.

```
(defcritic (deliberative*observes-opposite-of-actor-desire H)
  (in conditions current-conditions
    (desires ACTOR (REL SUBJ OBJ TRUTH)))
  (lisp OPPOSITE (if (eq TRUTH 'true) 'false 'true))
  (in hypotheses H
    (observes ACTOR (REL SUBJ OBJ OPPOSITE)))
  (=>))
```

Deliberative*Actor-Causes-Problem-For-Itself. In a given hypothesis, the actor takes an action, but this results in one of its present desires being undone.

```
(defcritic (deliberative*undoes-actor-desire H)
  (in conditions current-conditions
    (desires ACTOR (REL SUBJ OBJ TRUTH)))
  (lisp OPPOSITE (if (eq TRUTH 'true) 'false 'true))
  (in hypotheses H
    (sequential
      (does ACTOR (ACTION))
      (observes ACTOR (REL SUBJ OBJ OPPOSITE))))
  (=>))
```

Deliberative*Other-Actor-Undoes-Desire. In a given hypothesis, the other actor takes an action that undoes one of the actor's present desires.

```
(defcritic (deliberative*other-actor-undoes-desire H)
  (in conditions current-conditions
    (desires ACTOR (REL SUBJ OBJ TRUTH)))
  (lisp OPPOSITE (if (eq TRUTH 'true) 'false 'true))
  (in hypotheses H
    (sequential
      (does OTHER (ACTION))
      (observes ACTOR (REL SUBJ OBJ OPPOSITE))))
  (prolog (not (= ACTOR OTHER)))
  (=>))
```

Deliberative*Implication-Not-Inferred=>Add-Implication. In a given hypothesis, a first actor desires that a second actor take an action. In a narrative, when a first actor desired for a second actor to take an action, the first actor also desired for the second actor to believe that the first actor desired that the other actor take the action. Assert this implication into the given hypothesis. (Note that this deliberative critic does not depend on the existence of the narrative, as it does not draw any new elements from the narrative. One might think of the general knowledge captured by this critic—if someone wants someone else to do something, that someone else should know that the first person wants for them to do that thing—as "verified" by the provided narrative.)

```
(defcritic (deliberative*implication-not-inferred=>add-implication H N)
  (in hypotheses H
    (together
      (desires ACTOR (does OTHER (ACTION OTHER OBJECT)))
      (not (desires ACTOR
             (believes OTHER
               (desires ACTOR
                 (does OTHER (ACTION OTHER OBJECT))))))))
  (in narratives N
    (together
      (desires ACTOR2 (does OTHER2 (ACTION OTHER2 OBJECT)) [1])
      (desires ACTOR2
        (believes OTHER2
          (desires ACTOR2
            (does OTHER2 (ACTION OTHER2 OBJECT)))) [2]))
    (implies [1] [2]))
  (=>)
  (lisp NEW_H (extend-hypothesis H))
  (in hypotheses NEW_H
    (assert
      (desires ACTOR
        (believes OTHER
          (desires ACTOR
            (does OTHER (ACTION OTHER OBJECT)))) [[S]]]))
    (assert (subsit NEW_H [[S]]))))
```

Deliberative*Involves-Undesirable-Situation=>Prepend-Repair. There is a hypothesis in which an actor wishes for the other actor to believe that some state holds, and the other actor does not believe that this state holds. In the given narrative this undesirable state was eliminated by taking a particular action, and so that action is prepended to the hypothesis, causing the undesirable state to be eliminated. This critic is essentially engaging in a simple form of planning where the hypothesis (treated as a plan for action) is modified so that one of the unachieved goals of one of the actors is achieved.

```
(defcritic (deliberative*involves-undesirable-situation=>prepend-repair H N)
  (in hypotheses H
    (together
      (desires ACTOR (believes OTHER PROP))
      (not (believes OTHER PROP [S]))))
  (in narratives N
    (desires ACTOR2 (believes OTHER2 PROP2))
    (sequential
      (not (believes OTHER2 PROP2))
      (does ACTOR2 (ACTION ACTOR2 OBJECT) [CAUSE])
      (believes OTHER2 PROP2 [EFFECT]))
    (causes [CAUSE] [EFFECT]))
  (=>)
  (lisp NEW_H (extend-hypothesis H))
  (in hypotheses NEW_H
    (retract [S])
    (assert (does ACTOR (ACTION ACTOR OBJECT) [[S1]]))
    (assert (believes OTHER PROP [[S2]])
    (assert (subsit NEW_H [[S1]]))
    (assert (subsit NEW_H [[S2]]))
    (assert (causes [[S1]] [[S2]]))
    (assert (follows [[S2]] [[S2]]))))
```

3.5 Reflective Critics

As mentioned earlier, reasoning by procedurally applying deliberative critics, while efficient, is neither sound nor complete, which leads to errors such as incorrect inferences being made and correct inferences failing to be made. Thus the operation of deliberative critics is itself criticized by *reflective critics* that look at the trace of critic activity to identify problems with recent deliberations. Here are some examples of the kinds of criticisms that reflective critics can produce.

- Deliberation predicted that an action would succeed, but the action turned out to fail. This may have been because deliberation failed to apply a unit of knowledge that contained information about the required preconditions of that action.

- I have been working with another actor on a problem. Deliberation predicted that the other actor had correctly inferred my intent, but then the other actor took an action that made the situation worse. This may have been because the other actor did not actually have enough information earlier on to correctly infer my intent.

Why do we need reflective critics? Why not build inference systems that do not make these types of mistakes? The problem is that there is no known algorithm for efficiently making all useful and correct commonsense inferences given an uncertain problem context and a large commonsense knowledge base.[8] As a result, commonsense reasoning requires employing heuristic methods. The cost of using such methods is that they sometimes produce wrong inferences as well as neglecting to derive all potentially useful conclusions. In other words, they are unsound and incomplete. Even if inference engines were available that drew the best possible conclusions from the available evidence, the world is not fully observable and it changes, and so incorrect inferences are inevitable about the current state of affairs. In addition, in developing commonsense reasoning systems, there are problems other than just limitations in the efficiency and effectiveness of our inference methods—there may also be problems with the knowledge encoded in the commonsense knowledge bases used by those inference methods. Perhaps an item of knowledge, while true in many contexts, does not apply to the current situation.

Generally, there are many ways to go wrong while thinking! Reflective critics help cope with these limitations by recognizing problems when they occur, and helping explain and debug the source of the failure so that similar failures do not re-occur.[9,10]

---

[8] If we consider even the limited narrative representation used by EM-ONE, which is restricted to a fairly small vocabulary, the number of narratives that can be constructed by combining these vocabulary elements is enormous. To extend EM-ONE to broader domains, e.g. if it were to make use of the Cyc commonsense knowledge base, this problem would become far worse. On top of that, I have long considered using "ambiguous" representations where any given symbol could potentially be bound to a large number of potentially meanings, which I expect would make the situation exponentially worse. As a result, any approach to commonsense reasoning that seeks completeness, e.g. by engaging in some form of exhaustive search, is out of the question.

[9] Although, sometimes reflective critics identify failures that you really could not have done very much about. It's difficult to not brood about these failures just as much as about failures you could have done something about, because it's often not apparent that you couldn't have done anything about them until after you have further analyzed the episode. However, in this chapter I will try to focus on reflective critics that lead to some sort of corrective action.

[10] The reflective layer operates on the assumption that there are identifiable mistakes within the operation of the lower layers. This "debugging perspective" presumes a view of credit assignment that is more focused, local, and analytic. This is different from the conventional view in modern machine learning, where mistakes are incorporated into the assessment portion of a more broad, global, and somewhat holistic search for hypotheses that cover the positive examples but not the negative ones.

In this section I will focus on the description of reflective critics. In addition, I will briefly review the critics that populate the upper reflective layers of the EM-ONE architecture: self-reflective critics, self-conscious critics, and self-ideals critics.

3.5.1 Instrumenting critics for reflection

In CRITIC-L, when a critic is evaluated or applied, a trace is kept of its activation. If the critic is applied, any operations that are performed are recorded as well. Figure 3-4 shows an example of this trace. This trace keeps track of which critic calls which other critic, and what facts are asserted or retracted by critics when they are applied. E.g. line 54 states that the criticism T82449 (produced by the critic **meta*elaborate-hypotheses-to-infer-consequences**) results in the invocation of the critic **deliberative*known-action-consequence=>hypothesize-by-analogy**.

```
52: (calls T80780 (T82448 meta*brainstorm-to-infer-consequences))
53: (calls T82448 (T82449 meta*elaborate-hypotheses-to-infer-consequences))
54: (calls T82449
      (T82474 deliberative*unknown-action-consequence=>hypothesize-by-analogy
        seed-hypothesis hands-are-full))
55: (calls T82474
      (T82510 deliberative*unknown-action-consequence=>hypothesize-by-analogy
        seed-hypothesis attaching-stick))
56: (asserted-by
      (subsit N-N_TOPSIT_57020_82513 N-N_SIT_57022_82512)
      (T82510 deliberative*unknown-action-consequence=>hypothesize-by-analogy
        seed-hypothesis attaching-stick))
-------------------
70: (calls T82474
      (T83033 deliberative*unknown-action-consequence=>hypothesize-by-analogy
        seed-hypothesis successful-grasp))
71: (calls T83033
      (T83069 deliberative*unknown-action-consequence=>hypothesize-by-analogy
        seed-hypothesis interferes-with-other))
-------------------
85: (calls T83112
      (T83113 deliberative*unknown-action-consequence=>hypothesize-by-analogy
        H-82514 attaching-stick))
86: (asserted-by
      (subsit N-N_TOPSIT_57020_83116 N-N_SIT_57022_83115)
      (T83113 deliberative*unknown-action-consequence=>hypothesize-by-analogy
        H-82514 attaching-stick))
```

Figure 3-4. The reflective trace kept by CRITIC-L

Reflective critics match against this trace. Typically this reflective trace is not matched against directly, but instead there are special auxiliary predicates that make it more convenient to look at. Some of the convenience predicates that are available for reflecting on critic activity include those listed in Table 3-2.

Table 3-2. Reflective Predicate Types

Reflective Predicates	**Intended Reflective Predicate Meanings**
(asserted-by FACT CRITICISM)	FACT was asserted by CRITICISM instance
(ultimately-asserted-by FACT CRITICISM)	FACT was asserted by CRITICISM or its descendants
(asserts FACT)	FACT was asserted
(engages CRITIC)	critic of type CRITIC produced a criticism
(called-by C1 C2)	critic C1 was called by critic C2
(hypothesis-created-by HYP CRITICISM)	hypothesis HYP was asserted by CRITICISM instance
(opinion-changed-about R S O)	opinion about truth of predicate R(S,O) has changed
(narrative-not-used ACTION NARR)	NARR not involved in producing hypotheses with ACTION

Some of these reflective predicates match against the trace directly. Others are built on top of existing reflective predicates. For example, the **narrative-not-used** predicate is implemented as follows:

```
(defpattern (narrative-not-used ACTION NARR)
  (prolog (hypothesis-created-by H (C_ID (C_NAME C_H C_N))))
  (prolog (holds hypotheses H (type SIT ACTION)))
  (prolog (not (= NARR C_N))))
```

This reflective predicate recognizes that none of the deliberative critics involved in producing hypotheses involving an action ACTION made use of the narrative NARR. The reflective critics in the following section make use of this particular reflective predicate.

3.5.2 Examples of reflective critics

The following are examples of reflective critics used by EM-ONE.

Reflective*Action-Failed-To-Achieve-Effect*Neglected-Required-Precondition=>Append-Critic. Posits that we failed to predict that an action would fail because recent deliberations neglected to make use of a narrative that described one of the action's required preconditions. Modifies the metacritic responsible for those deliberations: the next time this metacritic is invoked, this narrative is used to verify that in hypotheses where this action is taken, the action is taken in circumstances where this precondition holds.

```
(defcritic (reflective*action-failed-to-achieve-effect
                    *neglected-required-precondition
                    =>append-critic N)
  (in conditions prior-conditions
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [A])
    (justifies [A] H)))
  (in hypotheses H
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [CAUSE])
    (observes ACTOR (REL SUBJ OBJ) [EFFECT])
    (causes [CAUSE] [EFFECT]))
  (in conditions current-conditions
    (observes ACTOR (not (REL SUBJ OBJ))))
  (in narratives N
    (sequential
      (together
        (observes ACTOR2 (REQUIRED S O))
        (does ACTOR2 (ACTION ACTOR2 S O))
        [1])
      (observes ACTOR2 (REL S O) [2]))
    (requirement [1] [2]))
  (prolog (narrative-not-used ACTION N))
  (=>)
  (in reflections recent-reflections
    (assert (missing-precondition REQUIRED SUBJ OBJ)))
  (lisp (append-critic
          'meta=>assess-hypotheses-to-infer-consequences
          `(assess-deliberative*preconditions-do-not-hold ,N ,H))))
```

Reflective*Partner-Not-Helping*Other-Failed-To-Infer-Goal=>Credit-Assignment. We wish for the other actor to take an action that is part of our joint plan, but we now realize that the other actor does not know that we have that plan, something we had assumed earlier. We posit that this conclusion had not been rejected earlier because earlier we did not use a narrative in which one actor does not infer the other's plan because it does not observe the other one taking an action from that plan.

```
(defcritic (reflective*partner-not-helping
                    *other-failed-to-infer-goal
                    =>credit-assignment N)
  (in conditions current-conditions
    (desires ACTOR (does OTHER (ACTION OTHER OBJ)) [S])
    (plans ACTOR PLAN)
    (justifies [A] PLAN))
  (in reflections recent-reflections
    (sequential
      (asserts ACTOR (desires ACTOR (believes OTHER (plans ACTOR PLAN))))
      (asserts ACTOR (believes ACTOR (believes OTHER (plans ACTOR PLAN))))
      (asserts ACTOR (believes ACTOR (not (believes OTHER (plans ACTOR PLAN)))))))
  (in narratives N
    (plans ACTOR2 PLAN2)
    (sequential
      (does ACTOR2 (ACTION2 ACTOR2))
      (not (observes OTHER2 (does ACTOR2 (ACTION2 ACTOR2)) [1]))
      (not (believes OTHER2 (plans ACTOR2 PLAN2) [2])))
    (dependency [1] [2]))
  (prolog (narrative-not-used ACTION N))
  (=>)
  (lisp (append-critic
         'meta=>assess-hypotheses-to-infer-motivation
         `(assess-deliberative*dependency-does-not-hold ,N ,H))))
```

3.6 Upper Reflective Critics

In addition to reactive, deliberative, and reflective critics, there are several higher level critics that are not presently implemented in EM-ONE, but which may be included in a future version of the system: self-reflective, self-conscious, and self-ideals critics.

3.6.1 Self-reflective critics

Self-reflective critics assess the current situation in terms of global characteristics of the system's knowledge, experience, and skills. Examples of self-reflective critics include the following:

Self-Reflective*Several-Known-Action-Failures. Given a proposed action, it is known that several instances of attempts to apply this action failed (i.e. its expected effects did not occur.)

```
(defcritic (self-reflective*several-known-action-failures)
  (in conditions current-conditions
    (intends ACTOR (ACTION ACTOR OBJECT)))
  (failed-at-action ACTION ACTOR1 OBJECT N1)
  (failed-at-action ACTION ACTOR2 OBJECT N2)
  (failed-at-action ACTION ACTOR3 OBJECT N3)
  (prolog (all-different N1 N2 N3))
  (=>))
```

Self-Reflective*Never-Taken-Action. Given a proposed action, the system has never before taken this type of action.

```
(defcritic (self-reflective*never-taken-action)
  (in conditions current-conditions
    (intends ACTOR (ACTION ACTOR OBJECT)))
  (not (in narratives N
         (does ACTOR2 (ACTION ACTOR2 OBJECT2))))
  (=>))
```

Self-Reflective*Lack-Knowledge-About-Event-Consequences. Given a proposed action, the system wishes to know its consequences, but there is no such knowledge in the narrative corpus.

```
(defcritic (self-reflective*lack-knowledge-about-event-consequences)
  (in conditions current-conditions
    (intends ACTOR (ACTION ACTOR OBJECT)))
  (not (in narratives N
         (does ACTOR2 (ACTION ACTOR2 OBJECT2) [1])
         (observes ACTOR2 (REL SUBJ OBJ) [2])
         (causes [1] [2])))
  (=>))
```

3.6.2 Self-conscious critics

Self-conscious critics assess the situation by comparing the system's activities and abilities with those of other actors. One such self-conscious critic would be "I failed at doing something that I believed the other agent believed I was good at."

3.6.3 Self-ideals critics

Self-ideals critics assess the situation based whether it is consistent with an system's "values," defined loosely as the system's highest priority general goals. One such self-

ideals critic would be the Golden Rule "do unto others as you would have them do unto you," in other words, criticize situations where the system is taking an action that results in a state for another actor that the system would consider undesirable if it were in the position of that other actor.

3.7 Assorted Mental Critics

This section lists some more mental critics that I have thought of but have not implemented yet as part of EM-ONE. Many of these are reflective critics were described in (Singh, 2003b), in which these critics were described both in English and in some cases also more precisely as declarative rules, using a small ontology of mental concepts (although a different representation from the one used in this thesis.)

No-Past-Success. This critic notices that a method that is currently being applied has never in the past succeeded.

Mistaken-Opportunity. This critic notices that the system was working on a solution, but the circumstances allowed it to quickly try something different. However, that failed, and unfortunately it also undid what progress it had made using the original method.

Another-Method-Available. This critic notices that the system was working on a solution using a difficult method, but then realizes that a simpler method had been available to it during that period.

Undoable-Negative-Side-Effect. This critic notices that the system took an action it should not have, because that action had a latent negative side effect that turned out to be undoable.

False-Subgoal. This critic notices that the system was distracted by a subgoal that did not help it achieve any of its more important goals.

Inaccessible-Resource. The system has assumed that a resource can be used to perform some function, but it turns out that the resource cannot be fully accessed.

Mistaken-Obstacle. The system expected certain objects or agents to be obstacles, but they turned out not to be and in fact they were helpful.

Wrong-Order. The system fails at several attempts to solve a problem. It realizes that if it had tried the last attempt first, it might have worked.

Undoable-Action. The system performed an action that could not be undone, even though it had expected it could be.

Undoable-Replacement. The system needs to replace a component. However, after it removes the original component it realizes that the new component is no longer available. The old component cannot be replaced and the system is left non-functional.

Too-Great-Risk. The action the system took was successful but it later realized that under the circumstances in which it was taken there was a fair chance it would have had a terrible outcome.

Misclassification. Several actions on an object failed in a row, and the system realized that it had classified the object being in one category when it fact it was in another.

Credit-To-Wrong-Action. Credit was given to a given action for producing an outcome, when it was really produced by another action.

Assumed-False-Preconditions. Actions are failing, and the system realizes that preconditions for those actions that had been assumed in fact did not hold.

Unable-To-Decide. Several methods seem to apply to the current problem, but a decision has failed to be made about which to select.

Wasted-Reasoning. While formulating a plan of action, the system realizes that the situation had changed and the problem had taken care of itself.

Lack-of-Experience. The system has had only a few experiences dealing with this problem.

Ignored-Relevant-Object. The system had expected a particular outcome from a given action, and in fact a different outcome had ensued because it interacted with an object that had previously not been noticed.

Transient-Conditions. The system had been depending on certain conditions to hold for a period of time, but in fact those conditions only held more briefly.

Misremembering. The system's memory of an event was revealed not to have been an accurate description of the original happening.

3.8 Summary

This chapter described many of the mental critics that are used in the EM-ONE system:

- **Reactive critics** accept observations from the outside world and propose or take actions in the outside world.

- **Deliberative critics** take hypotheses and narratives from the narrative corpus, and produce new and improved hypotheses that contain answers to questions that the system has, such as what would be the consequences of taking an action, or what motivations might underlie the actions of another actor.

- **Reflective critics** recognize problems in recent deliberations, and make repairs to the critics responsible.

- **Self-reflective critics** criticize the system's behavior based on models of limits in its knowledge and abilities.

- **Self-conscious critics** criticize the system's behavior by comparing itself to other actors and assessing their opinions of the system.

- **Self-ideals critics** criticize the system's behavior based on its consistency with top-level "values and ideals."

This collection is intended as a starting point and should not be considered "complete." Formulating new critics is for me an ongoing process, and one of my long-term goals with this research is to develop a comprehensive catalog of mental critics, as part of a more comprehensive theory of the processes and structures involved in ordinary commonsense thinking.

In the next chapter on meta-managerial critics I will discuss in more detail how these critics are coordinated.

Chapter 4

Meta-Management

> *Finally, we should note that in a creature with high intelligence one can expect to find a well-developed special model concerned with the creature's own problem-solving activity. In my view the key to any really advanced problem-solving technique must exploit some mechanism for planning—for breaking the problem into parts and shrewdly allocating the machine's effort and resources for the work ahead. This means the machine must have facilities for representing and analyzing its own goals and resources. One could hardly expect to find a useful way to merge this structure with that used for analyzing uncomplicated structures in the outer world, nor could one expect that anything much simpler would be of much power in analyzing the behavior of other creatures of the same character.*
>
> *— Marvin Minsky, in Matter, Mind, and Models (1965)*

The previous chapter described the reactive, deliberative, reflective, and upper-reflective mental critics that populate the EM-ONE architecture. This chapter describes how *meta-managerial critics* coordinate the activity of these mental critics.

4.1 Meta-Managerial Critics

In the EM-ONE architecture, there is great deal of freedom for critics to activate. Especially at the deliberative level, there are a tremendous number of inferences that are possible to make about a given situation, due to the great variety of deliberative critics that might match and commonsense narratives they might draw from. When these critics spawn new hypotheses or assessments of hypotheses, this only opens up many further avenues for deliberation. We need some way to guide and reign in all this mental activity.

To guide mental activity within EM-ONE, I have turned to the *critic-selector* model, a model of commonsense thinking proposed by Marvin Minsky (forthcoming) as a way to organize systems that make use of many forms of inference and knowledge representation. The central idea of the critic-selector model is that when the system encounters a problem, it brings to bear knowledge about what method of reasoning the system should employ to attack the problem. In EM-ONE such knowledge is captured by a special set of top-level critics, called *meta-managerial critics* (or *metacritics* for short.) Metacritics are concerned primarily with coordinating the activity of the layers of mental critics described in Chapter 3. Metacritics react to present conditions and progress that has been made so far to make decisions about which mental critics should be activated next, limiting the activation of mental critics to those that seem to be appropriate to the current overall problem-solving predicament. These metacritics operate at each "cognitive cycle" (after sensing the world but before taking actions upon the world) to decide what subset of mental critics should be active in the present moment—for example, whether the system should be acting, deliberating, or reflecting. Examples of metacritics include:

> Metacritic: We wish for the world to be a physical state different than it is.
> Way to Think: Seek a physical action that will make it so.
>
> Metacritic: There are several potential actions but it is unclear which is the best.
> Way to Think: Reject the actions that produce unacceptable consequences.
>
> Metacritic: We have taken an action but it has produced an unexpected outcome.
> Way to Think: Try to figure out why we failed to predict this outcome.
>
> Metacritic: My partner does not seem to have the same goals as I do.
> Way to Think: Try to explain how I failed to communicate my intent earlier on.

Thus, the activity in EM-ONE is not the result of a fixed algorithm or Soar-like decision cycle (Newell, 1990) where mental agents are invoked in some fixed order, but rather it is coordinated by metacritics that act by reacting to the present conditions and the progress so far. I think of these metacritics as forming a kind of expert system whose domain is the space of AI methods, and whose task is to select suitable AI methods for the present

problem situation. The term "meta-management" is due to Luc Beaudoin, a student of Aaron Sloman's who developed the idea theoretically in his PhD thesis (Beaudoin, 1994).

4.2 Networks of Metacritics and Mental Critics

Metacritics and mental critics are organized together into a tree-like hierarchical network as shown in Figure 4-1. The upper levels of the hierarchy are populated by metacritics and the lower levels of the hierarchy are populated by the mental critics from Chapter 3. When a metacritic within the network invoked, the network is traversed depth-first and left-to-right from that point, recursively invoking each critic that is encountered along the way. If a critic in the hierarchy fails to match, then its children are not invoked. In EM-ONE, there is a root metacritic **meta** that calls each available metacritic in turn.

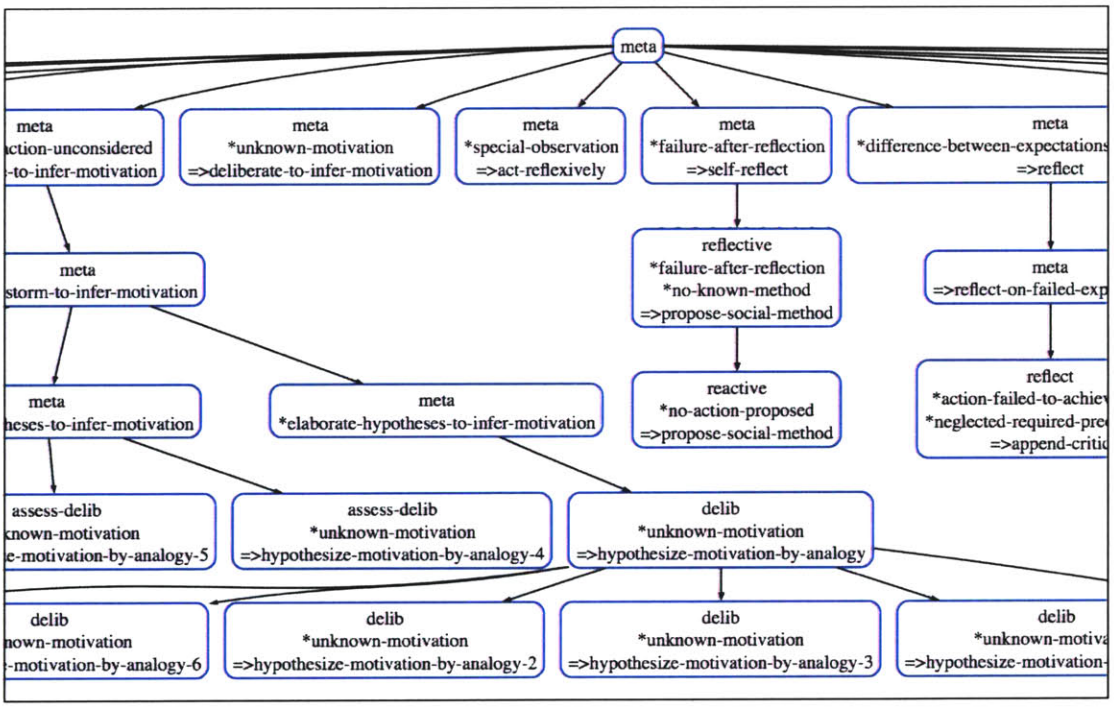


Figure 4-1. Subset of EM-ONE Critic Network

It may be useful to think of critic networks as resembling the behavior networks employed by the reactive planning community (e.g. Brooks, 1990), except—and this is a very important difference—that most of the "behaviors" are *mental* behaviors that

recognize the state and history of internal representations and processes, and act upon hypotheses within the mind and upon the code that produces this behavior.

4.3 Examples of Meta-Managerial Critics

Metacritics are represented in the same way as the mental critics described in Chapter 3. Here are several examples of metacritics:

Meta*Unachieved-Desire=>React. There is an unachieved desire. Propose actions to take by making an analogy to the narratives in the narrative corpus. Deliberate about the consequences of these proposed actions. If an action seems to be reasonable (has no negative consequences) then go ahead and take it in the world.

```
(defcritic (meta*unachieved-desire=>react)
  (in conditions current-conditions
    (observes ACTOR (REL SUBJ OBJ TRUTH))
    (desires ACTOR (observes ACTOR (REL SUBJ OBJ OPPOSITE))))
  (lisp OPPOSITE (if (eq TRUTH 'true) 'false 'true))
  (=>)
  (reactive*difference-between-conditions-and-desires=>propose-action-by-analogy)
  (meta*proposed-action-unconsidered=>deliberate-to-infer-consequences)
  (meta*proposed-action-seems-reasonable=>take-action))
```

Meta*No-Actions-Proposed=>Reformulate. The actor recently tried to come up with an action to achieve a desire, but no action was produced. There is a narrative that asserts that this desire is equivalent to another desire. Assert the other desire as one of the actor's present desires.

```
(defcritic (meta*no-actions-proposed=>reformulate N)
  (in reflections recent-reflections
    (engages ACTOR reactive*difference-between-conditions-and-desires
                  =>propose-action-by-analogy))
  (in conditions current-conditions
    (not (intends ACTOR ACTION))
    (desires ACTOR (REL1 SUBJ OBJ1))))
  (in narratives N
    (REL1 SUBJ OBJ1 [1])
    (REL2 SUBJ (REL3 SUBJ2 OBJ2) [2]))
    (jointly [1] [2]))
  (=>)
  (in conditions current-conditions
    (assert (desires ACTOR (REL2 SUBJ (REL3 SUBJ2 OBJ2)) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

Meta*Difference-Between-Expectations-and-Observations=>Reflect. Recognizes a proposed action did not achieve its expected effect. Reflect about why the action failed.

```
(defcritic (meta*difference-between-expectations-and-observations=>reflect)
  (in conditions prior-conditions
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [A])
    (justifies [A] H)))
  (in hypotheses H
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [CAUSE])
    (observes ACTOR (REL SUBJ OBJ) [EFFECT])
    (causes [CAUSE] [EFFECT]))
  (in conditions current-conditions
    (observes ACTOR (not (REL SUBJ OBJ))))
  (=>)
  (reflective*action-failed-to-achieve-effect
             *neglected-required-precondition
             =>append-critic))
```

Meta*Partner-Not-Helping=>Reflect. A first actor is solving a problem with a second actor. The first actor believes that the second actor desires a state that is opposite to a state desired by the first actor. This invokes reflective critics to consider whether the second actor is not helping because it failed to infer the goal of the first actor, and if so then (a) this should lead to a change in the deliberative machinery which caused the first actor to assume the other actor knew its intent, and (b) the first actor should immediately explicitly communicate its intent to the second actor.

```
(defcritic (meta*partner-not-helping=>reflect)
  (in conditions current-conditions
    (believes ACTOR (desires OTHER (REL SUBJ OBJ TRUTH)))
    (desires ACTOR (REL SUBJ OBJ OPPOSITE)))
  (lisp OPPOSITE (if (eq TRUTH 'true) 'false 'true)))
  (=>)
  (reflective*partner-not-helping*
              other-failed-to-infer-goal
              =>credit-assignment)
  (reflective*partner-not-helping*
              other-failed-to-infer-goal
              =>explicitly-communicate-intent))
```

Meta*Failed-At-Action*Conditions-Changed=>Try-Again. An actor took an action earlier that failed because there was a missing precondition. That precondition now seems to hold, so try taking the action again.

```
(defcritic (meta*failed-at-action*conditions-changed=>try-again)
  (in reflections recent-reflections
    (sequential
      (engages ACTOR meta*unachieved-desire=>react)
      (engages ACTOR reactive=>take-action)
      (engages ACTOR reflective*action-failed-to-achieve-effect
                              *neglected-required-precondition
                              =>append-critic))
    (missing-precondition PRECOND SUBJ OBJ))
  (in conditions current-conditions
    (observes ACTOR (PRECOND SUBJ OBJ))
  (=>)
  (meta*unachieved-desire=>react))
```

Meta*Want-To-Help-Other-Actor=>Play-Role. A first actor believes that the other actor is pursuing a plan. The first actor takes one of the actions from that plan in order to help the other one.

```
(defcritic (meta*want-to-help-other-actor=>play-role)
  (in conditions current-conditions
    (believes ACTOR (plans OTHER PLAN)))
  (action-from-plan PLAN ACTION SUBJ OBJ N)
  (=>)
  (in conditions current-conditions
    (assert (does ACTOR (ACTION ACTOR SUBJ OBJ) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

4.4 Ascending a Tower of Reflection

Metacritics can be organized to produce different patterns of mental activity. One such pattern that is used in EM-ONE results in an "ascending a tower of reflection" control system for selecting actions, producing inferences about problems with those actions, and reflecting upon that activity. That is, the layers can be organized so that the critics in higher layers respond to perceived or anticipated problems resulting from the activity of the critics in lower layers. This can be done by using metacritics to activate mental critics in the following order of operation:

1. **Reaction.** The system first invokes reactive critics that propose possible solutions to the current problem, by observing the current situation and comparing it against narratives from the EM-ONE narrative corpus. These reactive critics propose courses of action by matching narratives in which similar goals were achieved in similar conditions by taking particular courses of action. [11]

2. **Deliberation.** If actions are proposed by the reactive layer, deliberative critics are invoked to reason about the circumstances and consequences of those actions. The deliberative layer reasons by searching a space of hypothetical narratives starting from a seed hypothesis based on the present situation. This search is performed by iteratively applying deliberative critics that first complain about the present set of candidate hypotheses, and then proceed to spawn new hypotheses that try to

[11] The reactive layer can operate somewhat independently of the higher layers—even if there are no deliberative or reflective critics, the reactive critics can operate independently to act in the world. We may be able to measure the improvement in performance by turning on the deliberative and reflective layers. This will give us in the future a method with which to evaluate the architecture's performance.

improve upon those existing ones. The deliberative layer prefers hypotheses that are consistent with known narratives, that causally link the present situation to a clear future success or failure, that do not have major causal or explanatory gaps, that are relevant to the present context, that are rich with information, and that are internally consistent.

3. **Reflection.** The deliberative layer may encounter problems, such as producing incorrect predictions about the effects of actions. If such problems occur, then reflective critics are engaged to identify the source of these problems and modify the critics responsible for those errors so that they perform better in the future.

Because control is managed by metacritics, this is not the only style of control that is possible. One might instead choose to deliberate in advance of problems occurring (worrying), or spend no time reflecting upon recent deliberations, and so forth.

4.5 Chronic Mental Goals for Deliberation

This idea is not developed very much in EM-ONE, but the meta-managerial layer can be used to establish chronic questions and concerns that are useful to think about in *most* situations. Some of these are concerned with establishing details of the current context. Others are concerned with filling out details of likely past and future events. Still others are concerned with planning and anticipating for various outcomes that may occur. Yet others are concerned less with the details of the situation as it may have been or will be, but more general questions about the way the world is and our role within it. Such mental questions may drive some of the most ordinary commonsense thinking, to generate and pursue the answers to basic, chronic concerns. There are many possible question types, and some examples of these include:

- How did that object get there?
- What will happen next following this event?
- What would explain why this event occurred?
- What is the best thing for me to do now?

- What can I learn from this failure?
- What might go wrong while performing this action?
- What could be the negative consequences of taking this action?
- Why is that person taking that action?

In addition, there may be many types of mental goals that have less to do with the objects and events of the outside world, and more to do with the maintenance and improvement of the system's own cognitive machinery.

Each of these mental questions leads to other questions, and ultimately leads to ways of thinking that can attempt to address them. Meta-managerial critics are an appropriate place for such questions because they are at the top of the critic network and so can be regarded as persistent mental critics, ones that never turn off. When triggered, these metacritics trigger other critics that help more specifically to answer these types of questions. For example, if we wish to predict what might happen next in a situation, we may try to remember what happened next in a similar situation in the past. If we wish to learn from a failure, we may initiate a credit assignment process that traces back along the causal dependencies among recent events. And so forth. Sometimes the answers to these questions are immediately apparent. Other times some inference is needed, and often we cannot know the answers with absolute certainty. Surely, the decisions about whether these questions have been adequately answered are themselves subject to reflective thinking.

4.6 Summary

These metacritics are an initial step towards a richer ontology of the types of general predicaments that a commonsense AI system might face and methods for responding to those predicaments. While in this thesis I focus mainly on coordinating mental critics that do case-based reasoning, in the long run these metacritics could select between such varied techniques as different forms of statistical inference, logical theorem proving, the application of knowledge embedded in neural networks, as well as other styles of reasoning.

Chapter 5

Example Scenario

> *The ultimate goal of our research was to learn how to build entities capable of dealing intelligently with the full world in which humans operate. An important theoretical choice was between two directions. We could try to build a robot that would cope with the full complexity of the world encountered by a human child; like a baby the robot would begin by coping poorly, indeed very poorly, and gradually improve. The Blocks World exemplifies the other direction: designing a simplified world with which a robot could cope in a masterful way at a very early stage of development.*
>
> *– Seymour Papert, describing early research on the Blocks World at the MIT AI lab (Papert, 2004)*

In this chapter I will describe a detailed, implemented example that demonstrates how one might connect together the various architectural components described in the previous chapters to solve an ordinary commonsensical problem.

5.1 A Challenging Problem Domain

Seymour Papert has famously observed that "you can't think about thinking without thinking about thinking about something." Designing a cognitive architecture cannot be done entirely in the abstract. At some point it requires a concrete problem domain in which the design can be further developed, evaluated, and debugged. I argued in the introduction to this thesis that an architecture capable of commonsense reasoning should aim to achieve some competence within core mental realms including the physical, social, and mental realms. Given that little is known about how to build layered architectures for general broad-spectrum commonsense reasoning, I believe we should begin our explorations with problems set within simple domains requiring reasoning in primarily these core realms.

With this in mind, I have developed a problem domain that I believe is well suited to studying the core problems of commonsense reasoning. I have been have developing and debugging the core of the EM-ONE commonsense AI architecture in an artificial life environment called the *Roboverse*, a simulated world with realistic rigid-body physics populated by several actors whose actions are guided by the EM-ONE cognitive architecture.[12] These actors work together to build structures such as tables and chairs from simple, modular components such as sticks and boards. These components are tinkertoy-like in that they attach to one another at their corners and endpoints. The actors themselves are simulated robots; each possesses a synthetic perceptual system, and takes physical actions by controlling torques at simulated motors at its joints. They are vaguely humanoid in shape, each a human-like upper torso with a single arm, mounted on an inverted pendulum two-wheel base balanced by a PID servo.[13] Their arms end in spherical "hands" that allow a limited amount of manipulation of the environment; the hands act like magnets that can be turned on and off, attracting nearby objects, and when an object comes into contact with the hand, it is possible to establish a fixed joint between the hand and the object so that the robot can spin the object around to re-orient it. A screenshot of the Roboverse is shown in Figure 5-1.

[12] The Roboverse is a multi-platform robot simulation environment developed by myself and Bo Morgan.

[13] Perhaps in a future version they will be legged rather than wheeled. This would enable new and more human kinds of behaviors like climbing onto a chair or up a ladder to reach a high object.

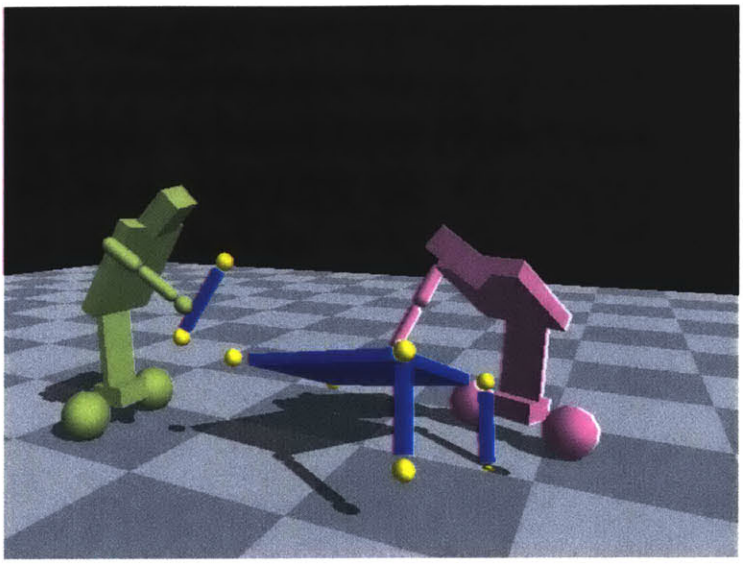


Figure 5-1. Working together to assemble a table from its constituent parts.

While this domain may seem sparse, its simplicity hides a great depth of issues. In particular, this world presents challenging problems within the physical, social, and mental realms. Because the world contains solid geometric objects that can collide and that behave according to Newton's laws, there are many problems that require competence at spatial and physical reasoning such as reasoning about the forces that must be applied to objects to move them about. Because there are several actors, the world emphasizes social problems in addition to physical ones, such as understanding conflicts between their goals and potential opportunities for cooperation. Because the world is quite open ended in the range of circumstances and problems that it admits, it is likely that variations in problem scenarios are distinct enough from known scenarios to cause the actors to make mistakes, and thus they will have to reflect on their own reasoning to understand those mistakes and avoid them next time. Any real world scenario involving several interacting agents is likely to involve some physical, social, and mental elements, and I believe there are many challenging cases of these problems within much more limited worlds than the real world. Once we are confident that we can build commonsense reasoning systems that function robustly in rich but limited domains such

as the Roboverse world, we can attempt to extend them to deal with broader ranges of problems using much broader arrays of commonsense knowledge.[14]

5.2 Connecting to the Virtual World Simulator

EM-ONE accesses the Roboverse virtual world via a collection of perceptual predicates for sensing the world and behavioral routines for influencing it. Perceptual predicates sense features such as the relative locations and orientations of objects, the actions being taken by other visible actors, as well as aspects of the actor's own state such as the direction it is currently facing. Behaviors can be initiated and terminated, and produce actions like moving, turning, reaching and grasping objects, looking in some direction, as well as more social actions like waving at the other actor. EM-ONE associates a frame type with each class of perceptual predicates and behavioral routines. Many of the perceptual predicates and behavioral routines I use are described Table 5-1, which is a subset of the ontology from Table 2-1 for which there are associated procedures in the virtual world simulator. The procedures in the virtual world simulator that implement these can be accessed from within EM-ONE. One can initiate a behavior by calling a lisp function, e.g. **(start-behavior (grasps :actor pink :object stick1))** initiates the action of grasping a stick, and **(stop-behavior (grasps :actor pink :object stick1))** terminates the action.

[14] Restricting the world in this way does not entirely bypass the need for large databases of commonsense knowledge, for to solve a wide range of problems in even the simple Roboverse world likely requires (at least) many thousands of elementary pieces of commonsense knowledge about space, time, physics, bodies, social interactions, object appearances, and so forth.

Table 5-1. Some of the available perceptual predicates and behavioral routines

Sensor Frames and Frame Slots	**Intended Sensory Frame Meanings**
is-visible-to :actor :object	OBJECT is visible to ACTOR.
is-touching :subject :object	the two objects are in contact
is-holding :actor :object	actor is grasping object and holding it
at-location :object :location	the object is located at place
has-speed :object :speed	the object is moving with speed
is-attached :subject :object	the two objects are attached
Behavior Frames and Frame Slots	**Intended Behavior Frame Meanings**
moves-to :actor :target	the actor moves next to the target
looks-at :actor :target	the actor looks at the target
grasps :actor :object	the actor grasps the object
releases :actor :object	the actor releases the object
attaches :actor :object :target	the actor attaches the object to the target

For accessing sensory information, there is a special lisp function **read-sensors** that calls all applicable perceptual predicates and asserts their values into the **current-conditions** database context. When **read-sensors** is called, the sensory facts in this context are first cleared and then freshly populated with frames representing what each robot can observe from its current location. The robots have direct access to the "visible" world state. This is defined as the state of the objects whose centers are within an expanding cone rooted at the head of the robot and whose axis is in the direction the head of the robot is oriented towards. This way, the robots are unaware of the world state that is out of view behind them. For all objects and actors visible to the robot, the robot has available to it the value of the predicates defining the objects' states, e.g. **(is-holding :actor pink :object stick1)**, and also the actions being performed by the actors, e.g. **(grasps :actor pink :object stick)**. In the simulator screenshot shown in Figure 5-1, a subset of the result of calling **read-sensors** is the set of observations shown in Table 5-2 below.

Table 5-2. Partial result of calling **read-sensors** for scene in Figure 5-1

```
(observes green (is-object-visible green stick1))
(observes green (is-robot-visible green pink))
(observes green (is-near-enough-to-reach green stick1))
(observes green (is-grasping-object green stick1))
(observes green (does green (grasps green stick1)))
(observes green (does pink (lifts pink board)))
...
(observes pink (not (is-object-visible pink stick1)))
(observes pink (not (is-robot-visible pink green)))
(observes pink (not (is-grasping-object pink stick1)))
(observes pink (does pink (lifts pink board)))
...
etc.
```

While this interface is a somewhat narrow and impoverished propositional channel through which EM-ONE can connect to the world, i.e. EM-ONE doesn't directly operate on the retinal images made available by the simulator or apply torques on the motors controlling its limbs, the right collection of perceptual predicates and behavioral routines can often make a simple propositional view of the world effective, at least for particular classes of problems. I chose this particular interface because it provided adequate ingredients for the robots to engage in physical tasks such as building simple structures from sticks and boards, and to engage in social tasks such as observing and saying things to the other robots. Many tasks involve some elements of each, e.g. one robot can hand a stick to the other one by combining moving towards it, reaching out while holding the stick, and releasing it when it observes that the other robot is holding the stick.

At the moment, the perceptual predicates and behavioral routines are implemented not in EM-ONE but in the simulator itself, so the reactive layer of EM-ONE is one layer of procedures separate from the lowest level sensors and effectors. The perceptual predicates are implemented largely by simple procedures that directly access world state—for example, **(is-near-to OBJ1 OBJ2)** is computed by directly measuring the distance between the two objects. The behavioral routines, such as moving towards objects and grasping them, are implemented using simple difference sensing processes and conditional reactive rules within the simulator system. While the simulator does its best to perform each of behaviors, they are not completely reliable because of the

difficulty of writing procedures to implement low-level motor control behaviors correctly. Just moving the robot from one position to another sometimes fails because the robot might overshoot that position, or while trying to grasp an object the arm might collide with an intermediate object. I decided to not spend more time than was absolutely necessary encoding these perceptual predicates and behavioral routines, because solving these problems was not central to my goal of demonstrating an implementation of EM-ONE. However, I am looking forward to a future version of the architecture that applies substantial commonsense spatial, physical, and bodily knowledge to the problem of perception and motor control itself.

5.3 Taking a Scenario-Based Approach

This virtual world, although simple compared to the real world, is still complex enough that large bodies of commonsense knowledge about the social, physical, and mental aspects of the world are needed to achieve generality—more knowledge than one person could implement today in the time span of a thesis project. Thus, rather than trying to build fully autonomous robots that live for extended periods in the virtual world while encountering a wide variety of problems, I have chosen to focus instead on the development of one particular scenario that requires only a very limited amount of knowledge. The scenario I will develop is the one presented at the very beginning of this thesis, which is shown again below in Figure 5-2, in which two creatures named Green and Pink work together to build a table from its component parts.

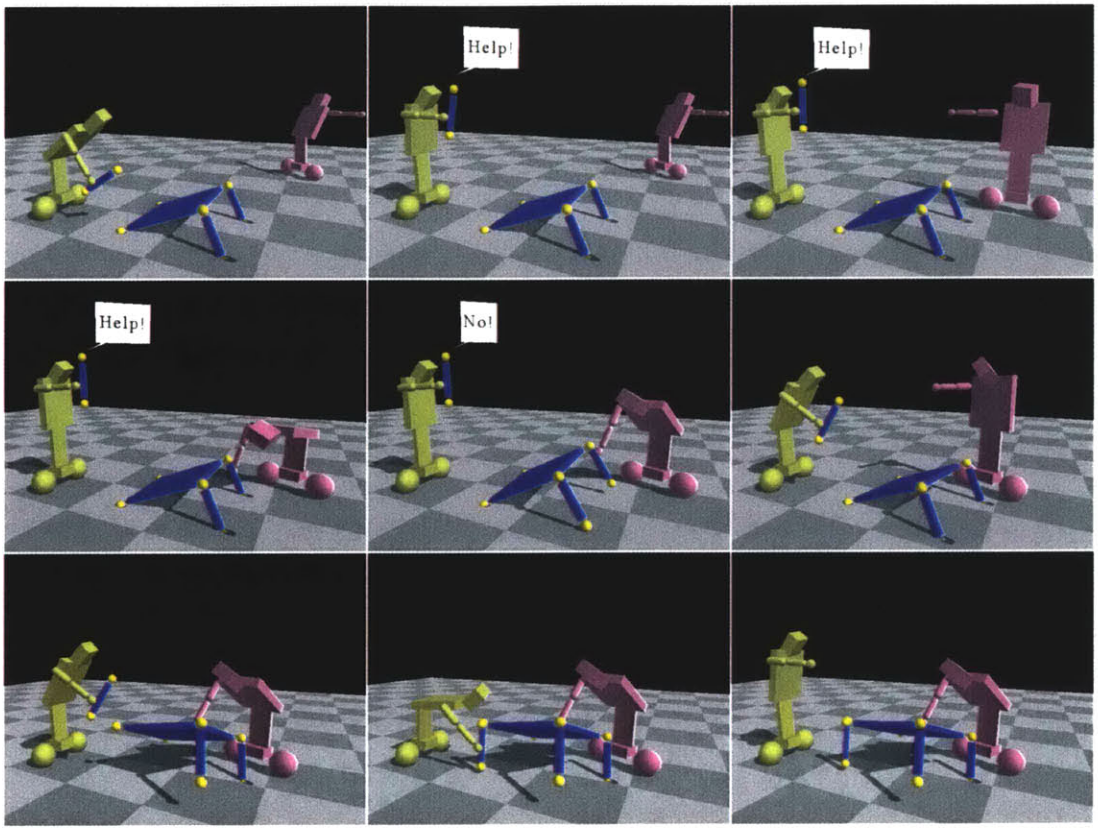


Figure 5-2. Building a table together

Again, here is the story: Green wants to build a table (perhaps, to place something on.) Green sees there is already a partly built table and realizes that it needs to attach more legs to complete the table. Green goes over and grabs a stick, and then goes over to the table. Green tries to attach the stick to the table but fails. Green quickly realizes that it needs help to insert the leg under the table, because Green only has one arm. Green calls over to Pink. Pink, who has been occupied with its own projects and has not been paying attention to Green until now, looks at Green holding a stick, and infers (mistakenly) that Green is trying to disassemble the partly built table. Pink comes over and starts to detach one of the table legs. Green realizes that Pink did not correctly infer Green's intent, and so complains. Green realizes that Pink did not see Green trying to attach the table leg. Green tries to attach the stick again to the table, this time with Pink watching. Pink now realizes that Green doesn't want to disassemble the table, but rather wants to complete

the table, and that Green expects Pink to hold up the table so that Green can attach the table leg it is holding. Pink holds up the table, and Green inserts the table leg underneath.

Stories like this one can be read as characterizing at a high level the series of physical, social, and mental actions that EM-ONE should take during the course of solving a given problem. In other words, they can be read as a "script" that captures not only what the actors in them should do, but also, what they should think. Thus, my goal with the following example is to show how a network of critics and narratives allows EM-ONE to "act out" the above scenario by engaging reactive, deliberative, and reflective processes operating across the physical, social, and mental realms.

5.4 Detailed Example: Completing a Table

In the following sections, I will describe how EM-ONE produces the above scenario. I have decomposed this scenario into the 10 scenes listed below. Each of these scenes demonstrates a few specific types of commonsense thinking.

1. Green wants to complete the table
2. Green thinks of attaching a leg to the table
3. Green tries and fails to attach a leg to the table
4. Green asks for Pink's help
5. Pink responds to Green's call for help
6. Pink infers (incorrectly) that Green wants to disassemble the table
7. Green recognizes that Pink failed to infer Green's real intention
8. Green communicates its intention to Pink
9. Pink infers Green's intention to add a leg to the table
10. Green attaches the table leg successfully

In the implemented version of this example, each of these scenes is produced by a separate EM-ONE critic network. These scenes are loosely coupled to the simulator and to each other. The EM-ONE critic networks do not access the Roboverse simulator directly. I manually supply the initial state of each scene based on a subset of the

observations the simulator produces for the scene, and on a subset of the relevant state that has persisted from the previous scene, including relevant parts of the reflective trace produced by the activity of critics in prior scenes. If at the end of the run of the critic network an action has been queued up to be taken by one of the robots, I manually feed that action to the simulator. In the next version of EM-ONE, I plan to connect it directly to the simulator so that the entire episode is generated in a single long run of a fully integrated EM-ONE critic network, but I found that it was much easier to develop the EM-ONE critic networks for the scenario by dividing it into smaller scenes in this manner. These critics networks were not intended to be highly general purpose—the critics and the narrative knowledge that are brought to bear were hand crafted for this particular scenario—but instead were intended to demonstrate the viability of employing a layered architecture for commonsense thinking based on the use of mental critics. Nevertheless, some generality is demonstrated by the fact that many of the same critics are used by the different scenes of the overall scenario.

1. Green wants to complete the table

Problem-Type: Reformulating a goal into a form for which a solution method is available.

Green starts off with the desire to build a table. This takes the form of a simple proposition asserted in the *current-conditions* database, stating that Green desires to observe a table.

```
(in conditions current-conditions
  (desires green (observes-class green table)
  (observes green (is-attached stick1 board))
  (observes green (is-attached stick2 board)))
```

The **meta*unachieved-desire=>react** critic invokes the reactive layer to propose a course of action to achieve this goal. This calls the **reactive*difference-between-conditions-and-desires=>propose-action-by-analogy** critic to propose an action. However, no action is proposed because there is no narrative where this particular goal,

described in precisely this way, is achieved. The **meta*no-action-proposed=>reformulate** critic then attempts to reformulate the goal.

```
(defcritic (meta*no-actions-proposed=>reformulate N)
  (in reflections recent-reflections
    (engages ACTOR reactive*difference-between-conditions-and-desires
                  =>propose-action-by-analogy))
  (in conditions current-conditions
    (not (intends ACTOR ACTION))
    (desires ACTOR (REL1 SUBJ OBJ1))))
  (in narratives N
    (REL1 SUBJ OBJ1 [1])
    (REL2 SUBJ (REL3 SUBJ2 OBJ2) [2]))
    (jointly [1] [2]))
  (=>)
  (in conditions current-conditions
    (assert (desires ACTOR (REL2 SUBJ (REL3 SUBJ2 OBJ2)) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

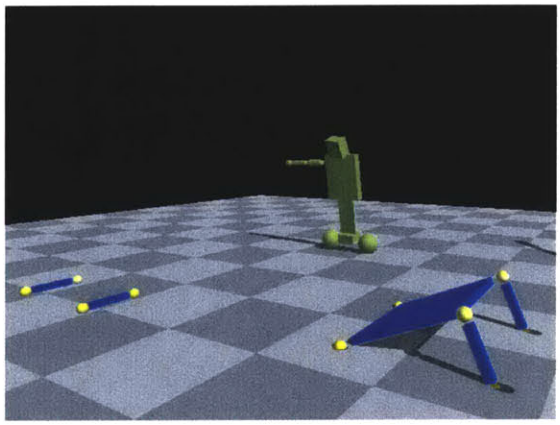
To do this it draws upon the **table-made-of-components** narrative, which equates observing a table with observing four sticks attached to a board.

```
(defnarrative table-made-of-components
  (together
    (observes-class pink table [1])
    (observes pink (is-attached stick1 board) [2])
    (observes pink (is-attached stick2 board) [3])
    (observes pink (is-attached stick3 board) [4])
    (observes pink (is-attached stick4 board) [5]))
  (jointly [1] [2])
  (jointly [1] [3])
  (jointly [1] [4])
  (jointly [1] [5]))
```

This deliberative critic reformulates the goal into the more concrete goal of observing a board that is supported by the four available sticks.

```
(in conditions current-conditions
  (desires green (observes-class green table))
  (observes green (is-attached stick1 board))
  (observes green (is-attached stick2 board))
  (desires green (observes green (is-attached stick1 board)))
  (desires green (observes green (is-attached stick2 board)))
  (desires green (observes green (is-attached stick3 board)))
  (desires green (observes green (is-attached stick4 board))))
```

The **meta*unachieved-desire=>react** critic then again invokes the reactive layer to propose a course of action. This time, it is recognized that there is a difference between the observed situation and the desired state that can be overcome by taking a particular action.



Step 1. Green wants to complete the table.

2. Green thinks of attaching a leg to the table

Problem-Type: Proposing an action to solve a problem.

The **reactive*difference-between-conditions-and-desires=>propose-action-by-analogy** critic recognizes that in the **attaching-stick** narrative, one of the problem differences was reduced by attaching a stick to the board.

```
(defnarrative attaching-stick
  (desires pink (is-attached stick board))
  (sequential
    (observes pink (not (is-attached stick board)))
    (does pink (attaches pink stick board) [1])
    (observes pink (is-attached stick board) [2]))
  (causes [1] [2]))
```

This reactive critic proposes attaching stick3 to the board as a course of action, resulting in the following intention asserted into the *current-conditions* database.

```
(in conditions current-conditions
  (intends green (attaches green stick3 board)))
```

The **meta*proposed-action-unconsidered=>deliberate-to-infer-consequences** critic sees that there is a intended course of action, but its consequences have not yet been considered by the deliberative layer. It establishes a seed hypothesis in the deliberative layer that includes the desire to build a table, the current observations about the situation, and the proposed action (asserted as one actually taken by Green, using the **does** frame, as opposed to **intends** frame.)

```
(in hypotheses seed-hypothesis
  (desires green (is-attached-to stick3 board))
  (observes green (not (is-attached-to stick3 board)))
  (does green (attaches green stick3 board)))
```

It then invokes the **meta=>brainstorm-to-infer-consequences** critic to cause the deliberative layer to begin work on making inferences from the seed hypothesis. For clarity of the following description, only a single iteration of brainstorming is engaged. This metacritic begins by invoking the **meta=>elaborate-hypotheses-to-infer-consequences** critic, which invokes a set of deliberative critics to respond to the seed hypothesis. Multiple narratives specific to this type of deliberation problem are brought to bear. In particular, the **deliberative*unknown-action-consequence=>hypothesize-by-analogy** critic complains that the action proposed in the seed hypothesis, to attach the stick to the board, has unknown consequences, and it sees that in the **attaching-stick** narrative, attaching a stick to a board causes the stick to be attached to the board. It thus generates a new hypothesis where after Green attaches the stick to the board, Green observes that the stick is attached to the board.

```
(in hypotheses H-19278
  (desires green (is-attached stick3 board))
  (sequential
    (observes green (not (is-attached stick3 board)))
    (does green (attaches stick3 board))
    (observes green (is-attached stick3 board))))
```

Other deliberative critics respond as well. Some of the additional potential consequences that are anticipated include (a) Green will no longer be holding the stick and (b) Pink might then undo Green's goal by detaching the stick from the board (which requires that Pink is near the stick, and so that is asserted as well.)

```
(in hypotheses H-19279
  (desires green (is-attached stick3 board))
  (sequential
    (observes green (not (is-attached stick3 board)))
    (does green (attaches stick3 board))
    (observes green (not (is-holding green stick3)))))

(in hypotheses H-19282
  (desires green (is-attached stick3 board))
  (observes green (is-near-to pink stick3))
  (sequential
    (observes green (not (is-attached stick3 board)))
    (does green (attaches stick3 board))
    (does pink (detaches stick3 board))
    (observes green (not (is-attached stick3 board)))))
```

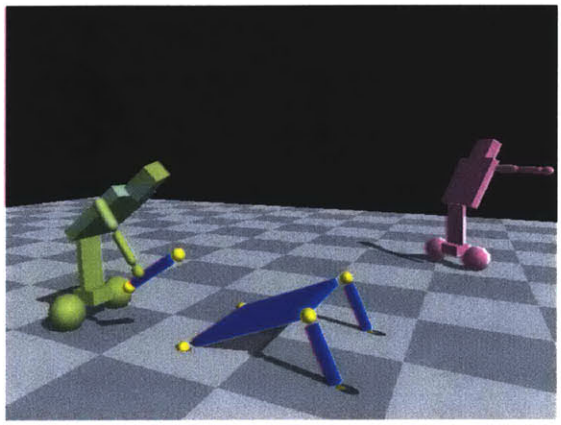
The next step of brainstorming is then taken. The **meta=>assess-hypotheses-to-infer-consequences** critic invokes deliberative critics to produce assessments of these generated hypotheses, in particular whether they are consistent with present observations and beliefs about the situation (whether they relevant to the current situation), consistent with known narratives (whether they are plausible), or achieve or undo goals (whether they are important.) In the above case, the hypothesis where Green is no longer holding the stick is criticized as irrelevant to current goals, and the hypothesis where Pink undoes Green's goal is criticized as inconsistent with observed conditions (since Pink is presently far from the table.)

The last step of brainstorming is then taken. The **meta=>filter-hypotheses** critic eliminates all but the top three least criticized hypotheses. The **meta*proposed-action-seems-reasonable=>take-action** critic sees that in the least criticized hypothesis the action seems to achieve the desire with no criticisms anticipated. The metacritic thus invokes the reactive layer to cause the action to be taken, justified by the highest ranking

hypothesis that was generated by the deliberative layer, by adding the following statements to the *current-conditions* database.

```
(does green (attaches green stick3 board) [ACTION])
(justifies [ACTION] H-19278)
```

This action is then fed to the simulator, resulting in the situation shown in Step 2 below.



Step 2. Green thinks of attaching a leg to the table.

3. Green tries and fails to attach a leg to the table

Problem-Type: An action fails to achieve its expected effect. Reflecting on what went wrong.

After the action is completed, a new set of observations is obtained from the simulator, and the formerly present conditions are asserted as now prior conditions.

```
(in conditions current-conditions
  (observes green (not (is-attached-to stick3 board)))

(in conditions prior-conditions
  (observes green (not (is-attached-to stick3 board)))
  (does green (attaches green stick3 board) [ACTION])
  (justifies [ACTION] H-19278))
```

The **meta*difference-between-expectations-and-observations=>reflect** critic checks to see if the hypothesis justifying the action just taken is inconsistent with the current

conditions. It complains that the prediction made by the hypothesis justifying the action did not turn out to occur, and recognizing that an action has failed to produce its expected effect, it invokes the reflective layer.

```
(defcritic (meta*difference-between-expectations-and-observations=>reflect)
  (in conditions prior-conditions
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [A])
    (justifies [A] H)))
  (in hypotheses H
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [CAUSE])
    (observes ACTOR (REL SUBJ OBJ) [EFFECT])
    (causes [CAUSE] [EFFECT]))
  (in conditions current-conditions
    (observes ACTOR (not (REL SUBJ OBJ))))
  (=>)
  (reflective*action-failed-to-achieve-effect
             *neglected-required-precondition
             =>append-critic))
```

The recent activity of the mental critics and their effects has been recorded as part of the ordinary operation of the CRITIC-L evaluator. Several reflective critics attempt to match against the recent reflective trace. The **reflective*action-failed-to-achieve-effect*neglected-required-precondition=>append-critic** critic matches, using the **narrative-not-used** reflective predicate.

```
(defcritic (reflective*action-failed-to-achieve-effect
                      *neglected-required-precondition
                      =>append-critic N)
  (in conditions prior-conditions
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [A])
    (justifies [A] H)))
  (in hypotheses H
    (does ACTOR (ACTION ACTOR SUBJ OBJ) [CAUSE])
    (observes ACTOR (REL SUBJ OBJ) [EFFECT])
    (causes [CAUSE] [EFFECT]))
  (in conditions current-conditions
    (observes ACTOR (not (REL SUBJ OBJ))))
  (in narratives N
    (sequential
      (together
        (observes ACTOR2 (REQUIRED S O))
        (does ACTOR2 (ACTION ACTOR2 S O))
        [1])
      (observes ACTOR2 (REL S O) [2]))
    (requirement [1] [2]))
  (prolog (narrative-not-used ACTION N))
  (=>)
  (in reflections recent-reflections
    (assert (missing-precondition REQUIRED SUBJ OBJ)))
  (lisp (append-critic
          'meta=>assess-hypotheses-to-infer-consequences
          `(assess-deliberative*preconditions-do-not-hold ,N ,H))))
```

This reflective critic posits that the deliberative layer did not consider all the required preconditions for the action, based on the **fails-to-attach-stick** narrative. In this narrative, the actor fails at attaching a stick to a board because the board did not have enough space beneath it for the stick.

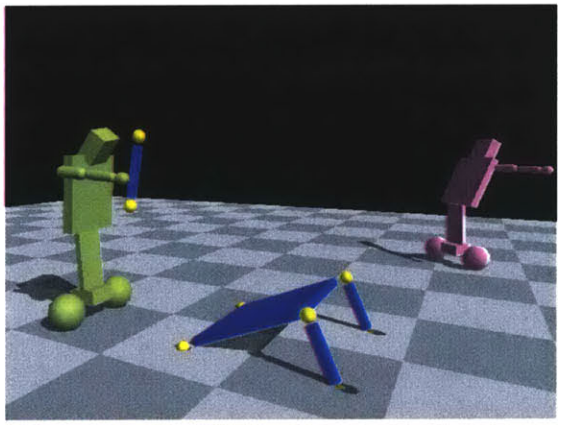
```
(defnarrative fails-to-attach-stick
  (sequential
    (together
      (observes blue (fits-beneath stick board))
      (does blue (attaches blue stick board))
      [CONDITIONS])
    (observes blue (is-attached stick board) [EFFECT]))
  (requirement [CONDITIONS] [EFFECT]))
```

This reflective critic looks at the trace of recent deliberation and recognizes that, earlier on in scene 2, Green predicted that it was possible to attach the stick to the table, and that none of the deliberative critics involved in that prediction made use of the dependency described in this narrative. When Green first attempted to attach the stick, it should have

predicted that the action would fail, but it did not infer at the time that there needed to be enough space to insert the stick. The reflective critic modifies the **meta=>assess-hypotheses-to-infer-consequences** critic so that it calls a deliberative critic that makes use of this narrative in making its predictions. Note that this is a case where an explanation for this type of failure had been available, but where the current network of critics was disposed to make this mistake because this knowledge was not brought to bear during deliberation. The reflective critic is thus making available to deliberation relevant knowledge that was already present but that had not been applied.

The **meta*reflective-modification=>redeliberate** critic, seeing that a reflective critic has made a change to the **meta=>assess-hypotheses-to-infer-consequences** critic, and that the problem still seems to exist, re-invokes the **meta*unachieved-desire=>react** critic to give it another chance to solve the problem.

However, with this modification, the action of attaching the stick to the board is rejected during the assessment phase of brainstorming—because a necessary precondition does not hold—and so this time no action is proposed.



Step 3. Green tries and fails to attach a leg to the table.

4. Green asks for Pink's help

Realizing that one cannot solve a problem alone. Coming up with a joint plan. Asking someone else for help.

In this scene, Green realizes that it cannot solve the problem alone, and so calls upon Pink to help.

Examining the reflective trace, the **meta*failure-after-reflection*no-known-method=>propose-social-method** critic complains that no action was proposed even after some reflection and additional deliberation.

```
(defcritic (reflective*failure-after-reflection
                      *no-known-method
                      =>propose-social-method)
  (in reflections recent-reflections
    (engages ACTOR meta*difference-between-expectations-and-observations=>reflect)
    (engages ACTOR meta*reflective-modification=>redeliberate))
  (in conditions current-conditions
    (not (intends ACTOR (ACTION ACTOR OBJECT)))
  (=>)
  (reactive*no-action-proposed=>propose-social-method))
```

Perhaps Green cannot solve this problem itself, and it suggests taking a social approach by engaging a reactive critic that employs narratives involving multiple actors. The **reactive*no-action-proposed=>propose-social-method** critic can now try to propose a method where Green and Pink work together.

```
(defcritic (reactive*no-action-proposed=>propose-social-method)
  (in narratives N
    (together
      (does ACTOR (ACTION1))
      (does OTHER (ACTION2))))
  (not (= ACTOR OTHER))
  (=>)
  (lisp H (assert-as-hypothesis N))
  (in conditions current-conditions
    (assert (plans ACTOR H [[SIT]]))
    (assert (subsit current-conditions [[SIT]]))))
```

This asserts the **completing-table-together** narrative as a modifiable hypothesis, which will serve as the initial joint plan.

```
(defnarrative completing-table-together
  (desires green (does pink (lifts pink board)))
  (sequential
    (does pink (lifts pink board))
    (does green (attaches green stick3 board))))
```

The **meta*proposed-plan-unconsidered=>deliberate-to-check-plan-conditions** critic engages the deliberative layer to look for problems with this proposed plan hypothesis. The plan hypothesis is elaborated by the **deliberative*implication-not-inferred=>add-implication** critic, using the **partner-does-not-know-desire** narrative, to conclude that Green desires that Pink believes that Green desires that Pink lifts the board.

```
(defnarrative partner-does-not-know-desire
  (together
    (desires green (does pink (lifts pink board)) [1])
    (desires green
      (believes pink (desires green (does pink (lifts pink board)))) [2]))
  (implies [1] [2]))
```

In the next phase of brainstorming, the **deliberative*involves-undesirable-situation=>prepend-repair** critic complains that Green has a desire involving Pink taking an action, but Pink does not know about that desire yet.

```
(defcritic (deliberative*involves-undesirable-situation=>prepend-repair H N)
  (in hypotheses H
    (together
      (desires ACTOR (believes OTHER PROP))
      (not (believes OTHER PROP [S]))))
  (in narratives N
    (desires ACTOR2 (believes OTHER2 PROP2))
    (sequential
      (not (believes OTHER2 PROP2))
      (does ACTOR2 (ACTION ACTOR2 OBJECT) [CAUSE])
      (believes OTHER2 PROP2 [EFFECT]))
    (causes [CAUSE] [EFFECT]))
  (=>)
  (lisp NEW_H (extend-hypothesis H))
  (in hypotheses NEW_H
    (retract [S])
    (assert (does ACTOR (ACTION ACTOR OBJECT) [[S1]]))
    (assert (believes OTHER PROP [[S2]])
    (assert (subsit NEW_H [[S1]]))
    (assert (subsit NEW_H [[S2]]))
    (assert (causes [[S1]] [[S2]]))
    (assert (follows [[S2]] [[S2]]))))
```

By turning to the **pink-helps-green** narrative, it resolves this criticism by proposing that Green ask Pink for help and letting Pink infer Green's goal. This results in the addition of the following step to the joint plan:

```
(does green (says green "Help!"))
```

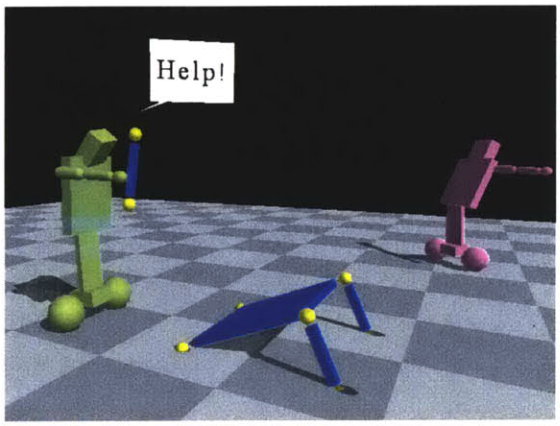
The final joint plan takes the following form:

```
(in hypotheses H-92393
  (desires green (does pink (lifts pink board)))
  (desires green (believes pink (desires green (does pink (lifts pink board)))))
  (does green (says green "Help!") [1])
  (believes pink (desires green (does pink (lifts pink board))) [2])
  (sequential
    (does pink (lifts pink board))
    (does green (attaches green stick3 board)))
  (causes [1] [2])
  (follows [1] [2]))
```

The **reactive*considered-plan-available=>take-next-action** critic begins to pursue this plan, adding the following statement to the *current-conditions* database.

```
(in conditions current-conditions
  (does green (says green "Help!")))
```

Note that in applying this particular plan, it is assumed that Pink will infer Green's goal of completing the table. As we shall see, this turns out not to be the case, because Pink did not see Green attempting to attach the leg to the table.



Step 4. Green asks for Pink's help.

5. Pink responds to Green's call for help

Problem-Type: Acting purely reactively.

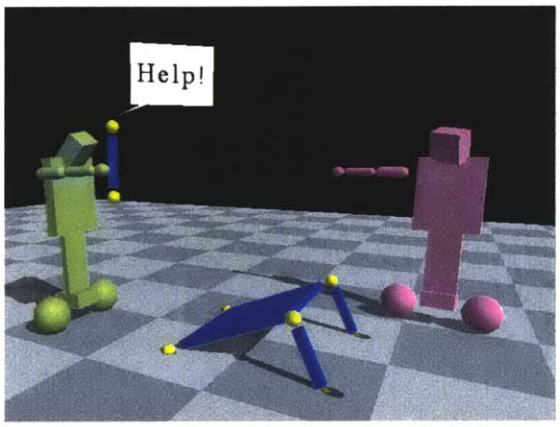
Let us switch now to Pink's point-of-view. One of Pink's observations is that Green asked for help.

```
(in conditions current-conditions
  (observes pink (says green "Help!")))
```

The **reactive*special-observation=>act-reflexively** critic responds to Green's call for help by employing the following narrative.

```
(defcritic (reactive*special-observation=>act-reflexively)
  (in conditions current-conditions
    (observes ACTOR (says OTHER "Help")))
  (prolog (not (= ACTOR OTHER)))
  (=>)
  (in conditions current-conditions
    (assert (does ACTOR (turns-toward ACTOR OTHER) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

Without additional deliberation, Pink turns towards Green.



Step 5. Pink responds to Green's call for help.

6. Pink infers (incorrectly) that Green wants to disassemble the table

Problem-Type: Participating in someone else's plan.

When Pink turns towards Green, this results in new observations.

```
(in conditions current-conditions
  (observes pink (is-holding green stick3)))
```

The **meta*unknown-motivation=>deliberate-to-infer-motivation** critic recognizes there is a belief about the state of the other actor, but the reason why the other actor is pursuing this goal is unknown, and so engages the deliberative layer to brainstorm about Green's motivations. The **deliberative*unknown-motivation=>infer-motivation-by-analogy** critic complains that the motivation of the actor has not been inferred, and

generates several hypotheses for why the actor is taking that action. After brainstorming the following two are the highest ranked hypotheses:

```
(in hypotheses H-22309
  (believes pink (plans green disassembles-table))
  (observes pink (is-holding green stick3)))

(in hypotheses H-22323
  (believes pink (plans green assembles-table))
  (observes pink (is-holding green stick3)))
```

The first hypothesis states that Green wants to disassemble a table, and the second that Green wants to assemble a table. These two hypotheses have an equal score (they received an equal number of criticisms), and it happens by chance that the first hypothesis is the one where Green wants to disassemble the table. This mistake is because to Pink there is no way to distinguish between these two hypotheses. At the end of brainstorming, the inferred motivation in this first hypothesis is asserted as a justified belief in the current conditions.

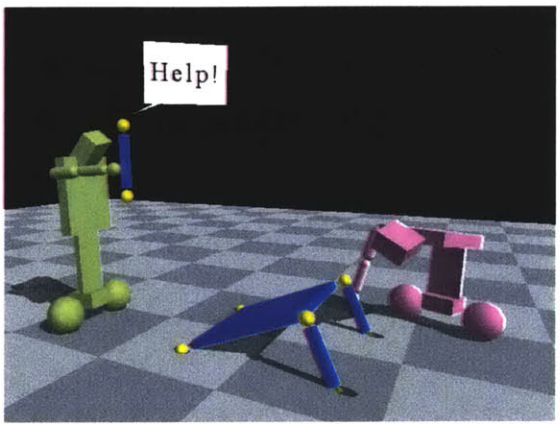
```
(in conditions current-conditions
  (believes pink (plans green disassemble-table) [1])
  (justifies [1] H-22309))
```

Pink now believes that Green wants to disassemble the table. The **meta*want-to-help-other-actor=>play-role** critic, using the **disassemble-table** narrative, causes Pink to intend to remove another one of the table legs.

```
(defcritic (meta*want-to-help-other-actor=>play-role)
  (in conditions current-conditions
    (believes ACTOR (plans OTHER PLAN)))
  (action-from-plan PLAN ACTION SUBJ OBJ N)
  (=>)
  (in conditions current-conditions
    (assert (does ACTOR (ACTION ACTOR SUBJ OBJ) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

This causes Pink to intend to detach one of the table legs, and so Pink begins to remove one of the attached table legs:

```
(in conditions current-conditions
  (does pink (detaches pink stick1 board)))
```



Step 6. Pink infers (incorrectly) that Green wants to disassemble the table.

7. Green recognizes that Pink failed to infer Green's real intention

Problem-Type: Inferring that someone else failed to infer your intention.

Let us return to Green's point of view. In this scene, Green infers that Pink had not inferred Green's intent. Initially, Green observes that Pink is attempting to detach stick1 from the board.

```
(in conditions current-conditions
  (observes green (does pink (detaches pink stick1 board))))
```

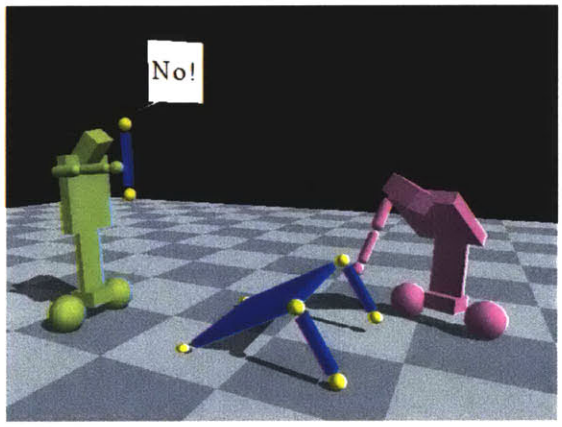
As Pink did in scene 6, the **meta*unknown-motivation=>deliberate-to-infer-motivation** critic sees that there is an observation of an action by another actor, so it engages the deliberative layer to brainstorm about the state of the other actor. The **deliberative*unknown-motivation=>hypothesize-motivation-by-analogy** critic, recognizing that no inference has been made about the motivation of the other actor, infers from the **disassemble-table** narrative that grasping a stick means wanting to disassemble the table, and posits a hypothesis where Pink has the goal of disassembling the table. After brainstorming, several such hypotheses are produced:

```
(in hypotheses H-25237
  (desires pink (disassembles pink table)))

(in hypotheses H-25283
  (desires pink (grasps pink stick)))
```

While Green does not have a single best candidate hypothesis, the **meta*partner-not-helping=>reflexive-complaint** critic complains that in all of these hypotheses, Pink is not helping Green with Green's goal, which causes the following action to be taken.

```
(in conditions current-conditions
  (does green (says green "No!")))
```



Step 7. Green recognizes that Pink failed to infer Green's real intention.

8. Green communicates its intention to Pink

Problem-Type: Reflecting on failing to communicate your intention. Clarifying your intent.

In the previous scene, Green inferred that Pink did not have the right goal in mind. In this scene, Green infers that Green did not properly communicate its goal earlier on, and decides to clarify its intention to assemble the table as opposed to disassemble the table.

Initially, the **meta*partner-not-helping=>reflect** critic is engaged, which engages the reflective layer.

```
(defcritic (meta*partner-not-helping=>reflect)
  (in conditions current-conditions
    (believes ACTOR (desires OTHER (REL SUBJ OBJ TRUTH)))
    (desires ACTOR (REL SUBJ OBJ OPPOSITE)))
  (lisp OPPOSITE (if (eq TRUTH 'true) 'false 'true)))
  (=>)
  (reflective*partner-not-helping*
              other-failed-to-infer-goal
              =>credit-assignment)
  (reflective*partner-not-helping*
              other-failed-to-infer-goal
              =>explicitly-communicate-intent))
```

This triggers the **reflective*partner-not-helping*other-failed-to-infer-goal=>credit-assignment** critic, which hypothesizes that Pink did not have enough information earlier on, by engaging a simple form of credit assignment that involves matching against the reflective trace that recorded the recent activity of the deliberative critics.

```
(defcritic (reflective*partner-not-helping*
                      *other-failed-to-infer-goal
                      =>credit-assignment N)
  (in conditions current-conditions
    (desires ACTOR (does OTHER (ACTION OTHER OBJ)) [S])
    (plans ACTOR PLAN)
    (justifies [A] PLAN))
  (in reflections recent-reflections
    (sequential
      (asserts ACTOR (desires ACTOR (believes OTHER (plans ACTOR PLAN))))
      (asserts ACTOR (believes ACTOR (believes OTHER (plans ACTOR PLAN))))
      (asserts ACTOR (believes ACTOR (believes OTHER (not (plans ACTOR PLAN)))))))
  (in narratives N
    (desires ACTOR2 (REL2 SUBJ2 OBJ2))
    (plans ACTOR2 PLAN2)
    (sequential
      (does ACTOR2 (ACTION2 ACTOR2))
      (not (observes OTHER2 (does ACTOR2 (ACTION2 ACTOR2))) [1])
      (believes OTHER2 (not (desires ACTOR2 (REL2 SUBJ2 OBJ2))) [2]))
    (dependency [1] [2]))
  (prolog (narrative-not-used ACTION N))
  (=>)
  (lisp (append-critic
         'meta=>assess-hypotheses-to-infer-motivation
         `(assess-deliberative*dependency-does-not-hold ,N ,H))))
```

It does this by using the **does-not-observe-actor-intent** narrative.

```
(defnarrative does-not-observe-actor-intent
  (desires green (is-attached stick board) [1])
  (plans green assemble-table)
  (sequential
    (does green (attaches green stick board) [2])
    (not (observes pink [2]) [3])
    (believes pink (not [1]) [4]))
  (dependency [3] [4]))
```

The **meta*partner-not-helping=>reflect** critic also triggers the **reflective*partner-not-helping*other-failed-to-infer-goal=>explicitly-communicate-intent** critic to engage the reactive response of explicitly communicating the goal.

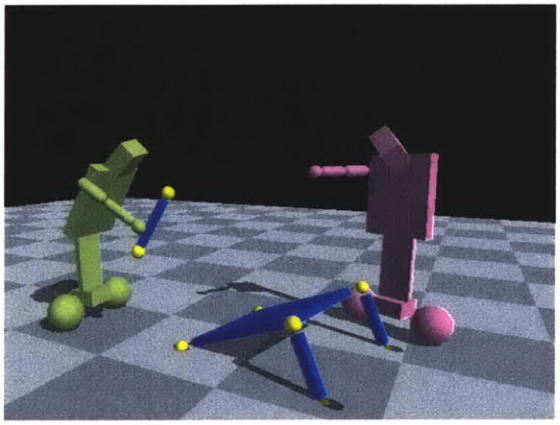
```
(defcritic (reflective*partner-not-helping
                    *other-failed-to-infer-goal
                    =>explicitly-communicate-intent)
  (in conditions current-conditions
    (desires ACTOR (REL SUBJ OBJ TRUTH)))
  (in reflections recent-reflections
    (sequence
      (asserts ACTOR (desires ACTOR (believes OTHER (plans ACTOR PLAN))))
      (asserts ACTOR (believes ACTOR (believes OTHER (plans ACTOR PLAN))))
      (asserts ACTOR (believes ACTOR (believes OTHER (not (plans ACTOR PLAN)))))))
  (=>)
  (reactive=>explicitly-communicate-intent))
```

This triggers the **reactive*explicitly-communicate-intent** critic to produce an action that communicates Green's intent to Pink. To communicate the desired goal and associated plan to Pink, Green takes one of the actions in that plan.

```
(defcritic (reactive=>explicitly-communicate-intent)
  (in conditions current-conditions
    (plans ACTOR PLAN)
    (observes ACTOR (is-visible-to OTHER ACTOR)))
  (prolog (not (= ACTOR OTHER)))
  (action-in-plan PLAN ACTION SUBJ OBJ)
  (=>)
  (in conditions current-conditions
    (assert (does ACTOR (ACTION ACTOR SUBJ OBJ) [[S]]))
    (assert (subsit current-conditions [[S]]))))
```

This causes Green to attempt to take a step of the plan.

```
(in conditions current-conditions
  (does green (attaches green stick3 board)))
```



Step 8. Green communicates its intention to Pink.

9. Pink infers Green's intention to add a leg to the table

Problem-Type: Inferring someone else's intention, and helping them.

Let us switch back to Pink's point of view.

```
(in conditions current-conditions
  (does green (attaches green stick3 board)))
```

The **meta*failed-at-inference=>redeliberate** critic sees that the other actor has taken a new action since it failed at inferring the other actor's motivation, and so it once again engages the **meta*unknown-motivation=>deliberate-to-infer-motivation** critic, as it did previously in scene 6.

```
(defcritic (meta*failed-at-inference=>redeliberate)
  (in reflections recent-reflections
    (sequential
      (engages ACTOR deliberative*unknown-motivation=>deliberate-to-infer-motivation)
      (asserts ACTOR (observes ACTOR (does OTHER (says OTHER "No!"))))
      (asserts ACTOR (observes ACTOR (does OTHER (ACTION))))))
  (=>)
  (deliberative*unknown-motivation=>deliberate-to-infer-motivation))
```

This time, because it has observed Green take an action from the **assembles-table** plan, it infers that Green actually wants to assemble the table rather than disassemble the table.

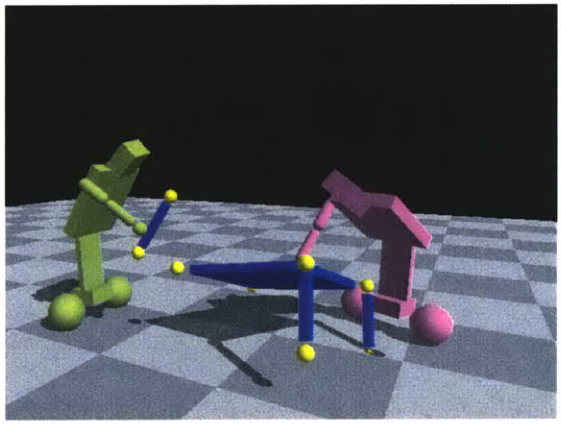
```
(in hypotheses H-24553
  (does green (attaches green stick3 board))
  (believes pink (plans green assembles-table)))
```

At the end of brainstorming, the inferred motivation is asserted as a justified belief in the current conditions.

```
(in conditions current-conditions
  (does green (attaches green stick3 board))
  (believes pink (plans green assembles-table) [1])
  (justifies [1] H-24553)
```

Pink now realizes that Green wants to assemble the table. The **meta*want-to-help-plan-of-other-actor=>play-role** metacritic, using the **assemble-table** narrative, causes Pink to intend to lift the table, making room for the leg to be inserted. The **reactive*take-action** reactive critic causes Pink to take this action.

```
(in conditions current-conditions
  (does pink (lifts pink board)))
```



Step 9. Pink infers Green's intention to add a leg to the table.

10. Green attaches the table leg successfully

Problem-Type: Acting according to one's plan.

Let us switch back to Green's point of view.

```
(in conditions current-conditions
  (observes green (fits-underneath stick3 board)))
```

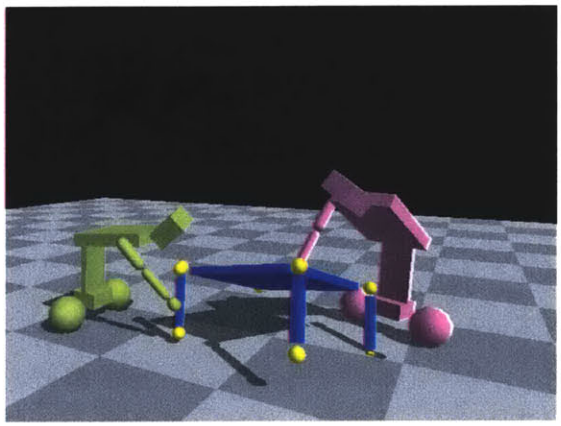
The **meta*failed-at-action*conditions-changed=>try-again** critic, seeing that a previously missing precondition for an action now holds, re-invokes the **meta*unachieved-desire=>react** critic (for a third time) to give it another chance to solve the problem.

```
(defcritic (meta*failed-at-action*conditions-changed=>try-again)
  (in reflections recent-reflections
    (sequential
      (engages ACTOR meta*unachieved-desire=>react)
      (engages ACTOR reactive=>take-action)
      (engages ACTOR reflective*action-failed-to-achieve-effect
                              *neglected-required-precondition
                              =>append-critic))
    (missing-precondition PRECOND SUBJ OBJ))
  (in conditions current-conditions
    (observes ACTOR (PRECOND SUBJ OBJ))
  (=>)
  (meta*unachieved-desire=>react))
```

The action of attaching the stick to the board is again proposed. This time, because its required preconditions hold and there seem to be no problematic consequences, the **meta*proposed-action-seems-reasonable=>take-action** critic invokes the reactive layer to cause the action to be taken, justified by the highest ranking hypothesis that was generated by the deliberative layer, adding the following statement to the *current-conditions* database.

```
(in conditions current-conditions
  (does green (attaches green stick3 board) [ACTION])
  (justifies [ACTION] H-34943))
```

This causes the action to be taken in the simulator, resulting in Green finally attaching the stick it is holding successfully to the board.



Step 10. Green attaches the table leg successfully.

5.5 Summary

This chapter provided a detailed example of how the elements described in the previous chapters can be combined to produce commonsensical behavior by a pair of actors working together to solve a problem. This example demonstrates EM-ONE as a kind of "cognitive programming language" in which one writes critic networks that engage forms of action, deliberation, and reflection. The particular EM-ONE critic networks that I developed cause two simulated robots to "act out" a desired episode, not by applying a fixed script of physical actions, but instead by applying a collection of mental critics that cause the robots to take actions as the result of reactive, deliberate, and reflective processes.

Chapter 6

Related Work

EM-ONE has its roots in the cognitive architectures that Marvin Minsky and Aaron Sloman have been developing in recent years. Their architectures have so many parallels that I sometimes call them jointly the Minsky-Sloman Architecture. I consider EM-ONE to be one example of an instance of the Minsky-Sloman Architecture. I have written more extensively about this connection in prior papers (e.g. McCarthy *et al.*, 2002; Singh & Minsky, 2003; Singh, 2003a; Minsky, Singh, & Sloman, 2004; Singh, Minsky, & Eslick, 2004; Singh & Minsky, 2005).

However, there were many other systems that inspired and influenced the design of EM-ONE, and this chapter relates EM-ONE to that prior work. To summarize in advance: (a) I first relate EM-ONE to the earliest work in AI concerned with critics, the GPS and HACKER programs. GPS can be seen as a collection of reactive critics, and HACKER as including a library of deliberative critics for debugging plans. (b) I then relate EM-ONE to Soar, which can be seen as based on four basic types of critics (which in Soar are called "impasses") that apply to the lowest-levels of processing in rule-based systems. (c) I describe how the narrative representation used in EM-ONE relates to the much richer representation used by the Cyc system. (d) I describe how Erik Mueller's ThoughtTreasure system inspired the many deliberative critics in EM-ONE as an example of a commonsense reasoning system that operates by applying a wide range of procedures, rather than by applying a small collection of general-purpose inference rules. (e) I describe John McCarthy's "mental situation calculus" as an inspiration for some of the representations that are used by EM-ONE's reflective critics. (f) I describe some of the similarities between EM-ONE and Belief-Desire-Intention architectures. (g) I discuss recent attempts to formalize human commonsense psychology. (h) I describe how EM-ONE can be regarded as a type of reflective case-based reasoning (CBR) system, and

relate it to some previous work in this area. (i) Finally, I describe how metacritics had their roots partly in Minsky's concept of the *Causal Diversity Matrix*.

6.1 General Problem Solver

One of the very earliest AI programs, Newell, Shaw, & Simon's (1960a) General Problem Solver (or "GPS") could be seen as a collection of critics. The critics of GPS were essentially reactive, in that they recognized differences between the system's goals and the present situation, and immediately took actions to reduce those differences; John McCarthy has referred to the GPS as a "symbolic servo-mechanism." GPS did not anticipate the consequences of those actions or otherwise consider their relative merits—it did not engage in the kinds of hypothesis-manipulation forms of deliberation that I discuss in this thesis. However, in (Newell, Shaw, & Simon, 1960b) they describe what could be the first AI system with a reflective level. The system they describe consisted of two separate GPS systems, one that operated on the base-level problem domain, and the second that operated on *the knowledge base of the first GPS.* This "second-order GPS" consisted of a set of critics that modified the critics of the first-order GPS so that their effects were more orthogonal and interfered with each other less. This is a form of reflection that EM-ONE does not presently engage in—reformulating the contents of its knowledge base in order to improve and accelerate inference. In EM-ONE, such an operation would probably be part of the self-reflective level, as it operates not so much on traces of recent deliberation but rather on the entire knowledge base of critics used by the system.

6.2 Hacker

The notion of a critic that embodies knowledge about bugs in programs was first explored by Gerald Sussman with the HACKER system he describes in his Ph.D. thesis (Sussman, 1973). This was the first program that operated by adapting known solutions to new situations, and that used a library of critics to debug these programs so that they would work in new situations. Sussman described a variety of critics, including a critic that resolved the now well-known Sussman Anomaly where a later plan-step undoes the effects of an earlier plan-step. Since Sussman's thesis, the use of such critics in

automated planning systems has become a relatively common technique. However, critics have seen little use in other areas of reasoning. In his thesis, Sussman suggests that critics are so important a type of knowledge that there should be a dedicated effort to develop a large catalog of such critics. Such a catalog has never been made, presumably because computer scientists have instead focused mainly on analyzing "correct" programs, rather than the partial solutions and "nearly correct" programs that our programs really are during most of the course of their development. In this thesis I have tried to take Sussman's suggestion seriously by beginning to catalog the kinds of critics that are useful for commonsense action, deliberation, and reflection.

6.3 Soar

Soar (Laird & Rosenbloom, 1996; Newell, 1990) is perhaps the best-known cognitive architecture in AI. The basic design of Soar is simple—it is essentially a rule-based system with special support for recognizing "impasses" in its problem solving. When Soar gets stuck, it immediately begins working on the new subproblem of resolving that impasse, applying its full deliberative capacity to the problem. Soar recognizes four types of impasses: (1) no-change, where no rule matches, (2) tie, where several rules match and we have no criteria for choosing between them, (3) conflict, where there is conflicting criteria for choosing between multiple rules, and (4) constraint-failure, where there are conflicting constraints concerned with choosing between multiple rules. These four impasses can be seen kinds of reflective critics, ones that would apply to any rule-based system where rules are selected in a two phase process in which rules are first proposed and then selected between by applying a separate set of preference rules.

Why does Soar only have four reflective critics while in this thesis I have discussed over a dozen? The reason it is easy to formulate reflective critics in EM-ONE is that critics in EM-ONE leave a trace of their operation, and every type of critical assessment of the system's performance that makes use of that trace is a candidate reflective critic. But in Soar, impasses are based only upon the number of ways that Soar's underlying rule-based system can get stuck, and so there are only a few types of this sort of failure. While there

are many prior conditions that could have led to those impasses, Soar does not have any special mechanisms for representing and identifying those causes of failure.

In addition, Soar does not have mechanisms for encoding "higher level" critics. Soar recognizes a few of the types of problems a rule-based system might run into at the lowest levels of rule selection and application. However, as one builds machinery by adding new rules and knowledge to Soar, this machinery will encounter entirely new sets of problems with that additional machinery, and Soar's built-in critics are of little help in identifying bugs in that higher-level machinery. Generally, as systems get more complicated by adding layers of abstraction, there are new ways things could go wrong at those higher layers of abstraction.

In my view, EM-ONE and Soar address orthogonal problems. Because Soar is a rule-based system, and the critics of EM-ONE are built using rules, I expect it would not be difficult to implement a version of EM-ONE using Soar as a substrate. EM-ONE focuses not on the underlying mechanisms for representing and applying facts and rules, but rather on the types of critics that a commonsense thinking machine should possess and their organization. Another way to understand this is that while Soar and EM-ONE are both cognitive architectures whose purpose is to support systems capable of human-level intelligence, they use two different senses of the term "architecture." Soar is based on the principle that to build human level intelligence we need to minimize the number of distinct mechanisms and representations that are used by our systems, and in Soar the term "architecture" refers to this minimum set of mechanisms (Soar Technology, 2002). In EM-ONE, I have used the term "architecture" to refer to the *structure and arrangement of commonsense knowledge and processes*. There is an analogue in biology. Every cell has an architecture that is at some level common across all cells and even to some extent across all living organisms. But to make up a person, these cells need to be organized into larger groups forming the various organs and systems of the body, each which possesses its own special ways to arrange cells. The brain, in particular, seems to be divided into hundreds of distinct centers. Soar makes few claims about the higher-

level organization of commonsense procedural and declarative knowledge, but the goal of EM-ONE is precisely to begin specifying that higher-level organization.

6.4 Cyc

The EM-ONE knowledge representation scheme was originally inspired by the Cyc upper level ontology. The Cyc project (Lenat, 1995) is the largest and most ambitious attempt to build a database of commonsense knowledge, and today Cyc contains over two million facts and rules about the everyday world. Its ontology is the most expressive presently available, and represents the state of the art in broad coverage formal knowledge representation. It includes a wide range of ideas about representing such matters as space, time, beliefs, goals, social relationships, physical constraints, as well as many other domains.

EM-ONE aspires to eventually support the use of large bodies of commonsense knowledge, and I considered using the Cyc ontology in EM-ONE. It was the only representation I had encountered that provided a sufficiently broad vocabulary to take the kinds of narratives and critics that I had been expressing pre-formally in English and express them in a sharper representation that a program could more easily work with. In the end, I decided to use my own, simpler scheme, although in the future I may try to implement a version of EM-ONE on top of the Cyc substrate. This should not be difficult because EM-ONE is built on top of a Lisp-based version of Prolog that makes use of Prolog's database and pattern matching machinery. Cyc provides this same functionality, but in many ways is more sophisticated because it has special purpose inference procedures for certain special types of commonsense inference (e.g. taxonomic and temporal inference are implemented with special-purpose algorithms.)

Some of the differences between EM-ONE and Cyc are:

(a) In EM-ONE there are a limited collection of frames and frame slots that let one describe physical, social, and mental situations. This simple representation scheme is far easier to learn and apply than the full Cyc ontology, and the cost in precision and

expressiveness trades off against Cyc's complexity. One of my goals at the outset of EM-ONE was to build a system that was reasonably easy for someone new to the project to understand, and I wanted to avoid putting them in the position of having to first learn the full Cyc ontology. In retrospect, the Cyc ontology is so well documented that it may have been a better decision to have selected an appropriate subset of Cyc. On the other hand, because the EM-ONE representation is completely frame-based, and because every frame is a reified entity that can be attached to a slots of other frames, it is especially natural to represent mental notions such as propositional attitudes (believes, desires, etc.) that relate actors to mental states, which is important for a system that does social reasoning and that can reflect upon itself and its own reasoning.

(b) The EM-ONE narrative corpus is based not on default rules but instead on narratives annotated by the causal and dependency relationships that exist between the elements of the narrative. These narratives are structured in a fairly uniform manner where each narrative is a set of situations whose constituents have simple causal and temporal interrelations. As I discussed in Chapter 2, I believe that narratives are generally a better way to represent commonsense knowledge than abstract logical rules, as is the convention in Cyc. Rather than reasoning by making deductions using commonsense rules, EM-ONE reasons by making analogies to commonsense narratives. While Cyc does have some knowledge in the form of stories and scripts, the bulk of the knowledge in Cyc is the form of general facts and rules.

(c) Cyc could be thought of as having a somewhat uniform deliberative layer, but like most present day inference systems it does not possess a reflective layer. Cyc possesses some knowledge about folk psychology and mental states, but it does not employ any of this knowledge to assess and debug its own functioning. Novel non-folk-psychological concepts about the mind, such as mental critics, do not appear. This seems to be because Cyc consists primarily of types of knowledge that AI practitioners are largely already familiar with, but theories about human commonsense psychology—ones that let us make detailed predictions about how

people learn, reason, and reflect—are still in their infancy, so there was little existing research for the developers of Cyc to draw upon.

One additional problem with attempting to build upon the Cyc inference engine is that I would like to make use of an underlying matching system that is more flexible than traditional symbolic unification. In the next chapter I will describe some ideas about how this might be done, using a class of techniques we have been calling "Panalogy."

6.5 ThoughtTreasure

One of the early influences on EM-ONE was Erik Mueller's ThoughtTreasure system (Mueller, 1998). ThoughtTreasure is a *tour de force* story understanding system that uses multiple representations, multiple methods of inference, and a substantial commonsense knowledge base possessing on the order of 100,000 items. ThoughtTreasure includes a semantic parser, a natural language generator, and most importantly a wide collection of "understanding agents" that make different inferences about the sentences it reads. When ThoughtTreasure is presented with the next sentence of a story, it simulates the world as it was understood previously up until that new story step, filling in the steps between using commonsense reasoning. (Mueller, 1999) summarizes some of the difficulties faced in building the ThoughtTreasure system. As ThoughtTreasure grew to acquire more resources—more representations, more methods of reasoning, and so forth—it eventually became too difficult to understand and to further improve.[15]

EM-ONE resembles ThoughtTreasure in that it produces commonsense reasoning using not a single inference procedure, but rather a large collection of agents that each makes specific types of commonsense inferences. EM-ONE has a somewhat more uniform architecture than does ThoughtTreasure, based on the use of critics and narratives. However, that uniformity was to some extent the result of wanting to present a simplified version of the Emotion Machine architecture for pedagogical purposes. I suspect that

[15] Erik Mueller likes to quote from "Spock's Brain," an episode of the original Star Trek series in which Dr. McCoy heroically attempts to reconnect Spock's brain to his body: "I'm trying to thread a needle with a sledge hammer. What am I supposed to do? I can't remember. I don't remember."

extending EM-ONE to support more types of representations and forms of inference would soon break this pleasant uniformity.

More importantly, ThoughtTreasure does not use reflective critics that assess its own deliberation processes. It seems to me that a good test of the scalability of EM-ONE would be to try to reimplement ThoughtTreasure or a similar story understanding system using critic-based control structures. In fact, one of the motivations for EM-ONE was to develop an architecture in which a system like ThoughtTreasure could be built, but where by adding reflective processes it could assess and repair some of its own reasoning processes, or at minimum, aid a programmer in its development.

6.6 Mental Situation Calculus

One of the inspirations for the representation of reflective critics in EM-ONE was a paper by John McCarthy's about consciousness (McCarthy, 1995). In that paper McCarthy suggests the development of a "mental situation calculus," one that involves predicates and axioms for explicitly representing such concepts as knowing, forgetting, believing, and other mentalistic notions. McCarthy sketches the beginnings of a formal theory of such a mental situation calculus, but leaves most of the details for others to work out. He explicitly observes the potential of reflective critics (although he does not use that particular term): "A robot should be able to wish that it had acted differently from the way it has done. A mental example is that the robot may have taken too long to solve a problem and might wish that it had thought of the solution immediately. This will cause it to think about how it might solve such problems in the future with less computation." He also sees the potential of self-reflective critics: "A human can wish that his motivations and goals were different from what he observes them to be. It would seem that a program with such a wish could just change its goals. However, it may not be so simple if different subgoals each gives rise to wishes, e.g. that the other subgoals were different." However, since McCarthy's paper there has not been very much work on formalizing concepts like regretting and wishing.

6.7 Belief-Desire-Intention Architectures

I did not draw very much upon this literature while developing the ideas in this thesis, but Belief-Desire-Intention (BDI) architectures (Rao & Georgeff, 1995) are a well-known framework for controlling the actions of agents. BDI architectures are centered around the explicit representation and manipulation of propositional attitudes such as beliefs, desires, and intentions. Many practitioners of BDI architectures develop practical agent systems based on formalizations of the relationships between these propositional attitudes. It seems to be a common research practice to contribute an axiomatization of the relationship between these propositional attitudes, although there is no single, agreed upon way to do so within the BDI community. Generally, researchers have observed that these concepts are subtler than one might realize at first examination. Examples of BDI axioms include such default rules as:

- If an agent believes that A implies B and the agent desires A, then the agent desires B
- If an agent desires to achieve A, it does not believe that not(A) is a certainty.
- If an agent intends to do A, then it does not believe that it will not do A.
- If an agent intends to do A, it does not necessarily intend all of the side effects of A.

See Bratman (1987), Cohen & Levesque (1990), and Rao & Georgeff (1991) for more such examples. The multiagent case for joint planning has been examined by Grosz & Sidner (1990), in which they present their SHAREDPLANS formalism, with a focus on dialogue in collaboration. Grosz & Kraus (1996) provide a fuller axiomatization of SHAREDPLANS. Pollock (1990) posits that we need to represent the "having of a plan" in addition to a plan itself, and this can be done by representing them as structured collections of beliefs and intentions.

6.8 Formal Theories of Commonsense Psychology

EM-ONE captures some BDI-like axioms within its commonsense narratives. However, achieving human level commonsense reasoning about mental activity will likely require a broader collection of mental concepts, for example, representations of how memory

works, the limits of perception, types of difficulties in reasoning, and so on. Compared to issues such as representing space, time, and physical objects, relatively little work has been done finding ways to represent the content of ordinary human psychology.

The most comprehensive outline I have run across of the things people do when thinking is Andrew Gordon's catalog of types of mental strategies used in commonsense planning (Gordon, 2004). More recently, Gordon and Hobbs have been attempting to represent some aspects of these mental strategies as formal logical theories (Gordon & Hobbs, 2004). Gordon is also taking a third approach, which is searching for instances of mental concepts in natural language corpora (Gordon & Nair, 2004). While the former two approaches to formalizing mental concepts seem to have been the result of a long introspective process, supplemented with considering what mental notions have been explored in the existing AI literature, this latter approach is more empirical, drawing on how people talk about psychological concepts in natural language.

I am excited about this line of work because it may be possible to use such techniques to infer procedural knowledge about how to think by reading natural language stories, which just as often describe mental activities as they do physical activities. In addition to learning from the mental activities of story characters, one may be able to learn patterns of reasoning from texts where the authors are explicitly discussing and explaining ideas, where arguments are put forth, developed, and rejected. EM-ONE's hand-crafted collection of critics and narratives could eventually be supplemented by more free form representations of mental processes.

6.9 Introspective Case-Based Reasoning

There have been several attempts to incorporate reflection into case-based reasoning systems resembling the way that EM-ONE does reflection. In particular, Cox and Ram (Cox, 1997; Cox & Ram, 1999) developed Meta-AQUA, an elaborate case-based explanation system that could diagnose its reasoning failures. The Meta-AQUA system uses a self-model to explain how and why its story-understanding component generated faulty explanations. Cox & Ram call such explanations *meta-explanation patterns* (Meta-

XPs), which are represented as directed graphs describing the causal relationships between mental states and processes. Meta-XPs resemble the antecedent side of the reflective critics of EM-ONE. The effect of Meta-XPs is different than reflective critics, however, in that Meta-XPs cause the system to make changes to its store of background knowledge, whereas reflective critics modify the procedures of other critics.

A second example is the ROBBIE system by Fox & Leake (1995), which incorporates introspection into a case-based planning system to give it the capacity to improve the way it indexes its cases. The ROBBIE system uses introspective reasoning to monitor the retrieval process of a case-based planner in order to detect retrieval of inappropriate cases. When retrieval problems are detected, it explains the source of the problems and uses those explanations to determine new indices to use for future case retrieval. It makes of use four case-based reasoning critics: (a) case memory lacks a required case, (b) reasoner fails to retrieve a relevant case, (c) reasoner retrieves an irrelevant case, and (d) reasoner improperly applies a retrieved case. This is similar in its effects to EM-ONE's reflective critics modifying metacritics to produce additional types of inferences involving new narratives, although reflective critics are intended to cover a broader range of self-repairs than repairs to the way narrative knowledge is retrieved.

6.10 Causal Diversity

The concept of meta-managerial critics described in this thesis derives partly from Minsky's *Causal Diversity Matrix* (Minsky, 1992), shown in Figure 6-1, a meta-theory of AI that suggests when to apply different AI techniques. Here, each problem-solving method, such as analogical reasoning, logical theorem proving, and statistical inference, is assessed in terms of its competence at dealing with problem domains with different causal structures.

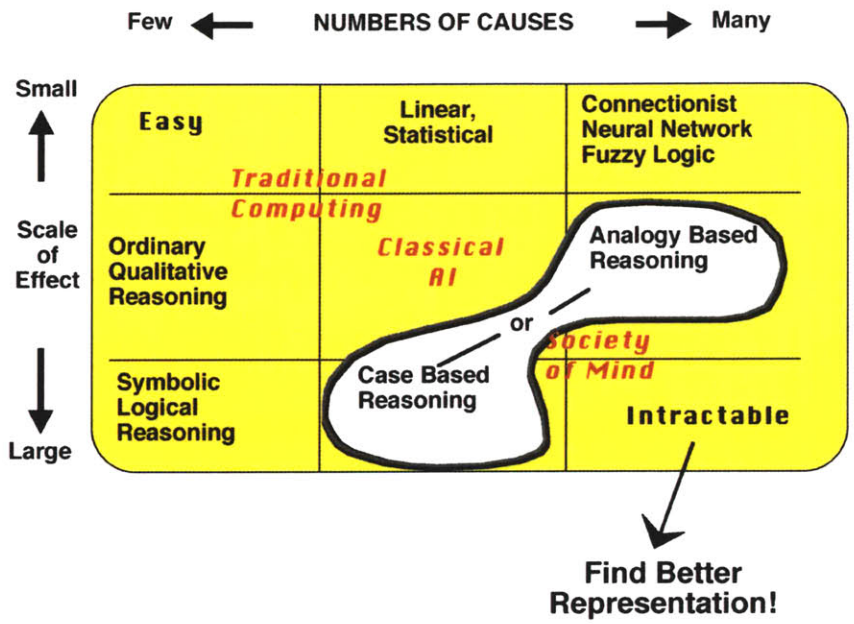


Figure 6-1. The causal diversity matrix

(Diagram from Minsky's *Future of AI Technology* (Minsky, 1992))

Figure 6-1 can be read as follows. Statistical inference is often useful for situations that are affected by many different matched causal components, but where each contributes only slightly to the final phenomenon. A good example of such a problem-type is visual texture classification, e.g. determining whether a region in an image is a patch of skin or a fragment of a cloud. This can be done by summing the contributions of many small pieces of evidence such as the individual pixels of the texture. No one pixel is terribly important, but *en masse* they determine the classification. Formal logic, on the other hand, works well on problems where there are relatively few causal components, but which are arranged in intricate structures sensitive to the slightest disturbance or inconsistency. An example of such a problem-type is verifying the correctness of a computer program, whose behavior can be changed completely by modifying a single bit of its code. Case-based and analogical reasoning lie between these extremes, matched to problems where there are a moderate number of causal components each with a modest amount of influence. Many common sense domains, such as human social reasoning, may fall into this category. Such problems may involve knowledge too difficult to

formalize as a small set of logical axioms, or too difficult to acquire enough data about to train an adequate statistical model.

It is true that many of these techniques have worked well outside of the regimes suggested by this causal diversity matrix. For example, statistical methods have found application in realms where previously rule-based methods were the norm, such as in the syntactic parsing of natural language text. However, we need a richer heuristic theory of when to apply different AI techniques, and this causal diversity matrix could be an initial step toward that. We need to further develop and extend such theories to include the entire range of AI methods that have been developed, so that we can more systematically exploit the advantages of particular techniques.

In the long run I hope that the use of metacritics will lead to improved AI systems that can make use of multiple different AI techniques in a single system, and further research into how to build systems that are populated with many types of more managerial agents whose area of expertise is not things in the outside world but rather the processes of thinking themselves—which includes for example such things as methods of inference, techniques for learning, modes of representation, manners of emotion, and presumably many other types of phenomena that we have no good words for.

Chapter 7

Future Work

There are many new features I would like to include in a future EM-TWO system. This chapter imagines possible next steps for EM-ONE.

7.1 Expanding the Catalog of Mental Critics

I am sure I have barely scratched the surface in identifying types of mental critics. While this thesis has taken a preliminary step towards the catalog of critics that Sussman advocated in his PhD thesis, there are as many critics as there are problems in the world—in other words, there is still a long way to go! At the very least, I hope to develop further the types of critics that are relevant within simple commonsense domains like the kind I studied in this thesis, as these are hopefully applicable to other domains where physical, social, and mental errors may occur.

7.2 Generalized Matching and Analogy-Making

In their present implementation, critics only match particular forms of knowledge. For example, a critic may match a situation that is represented as at least two relations between two objects, but would fail to match similar situations where there was only a single such relation. In the EM-ONE system, for many of the types of critics described in this thesis, there actually exist a family of critics that match slightly different structural forms of the same types of conditions. In the long run, a more flexible matching scheme should be employed where entire situations can be compared to one another in a single general operation (perhaps by computing whether one situation's description subsumes the other.)

In addition, the critics of EM-ONE engage in exact matching, and do not make partial analogies in the sense of analyzing similarities and differences between the source and target situations between compared, and assessing the importance of these differences when deciding whether to project new relations onto the target description. In EM-TWO, I would like to use a form of analogy where partial matching is the norm, and where the similarities and differences encountered during comparisons are recorded so that reflective critics can analyze them later if the critic results in failure. In other words, the matching process should be made available to reflection.

7.3 Heterogeneous Reasoning Modules

In EM-ONE, inferencing is done by simple graph matching and manipulation operations, and is of a rather uniform character. An alternative approach is to allow inferencing to be done by a diversity of specialized modules—e.g. modules for Bayesian inference on different network topologies, neural network propagation, resolution-based theorem proving, etc.—that are instrumented so that they can, at least partly, be reflected upon and improved. In such a system a critic might call upon an entirely separate inference engine for solving large temporal reasoning problems quickly, or matching shapes within a diagram, but where that inference engine keeps a partial trace of its activity so that specialized reflective critics can later reflect upon its mistakes. This approach would allow for a fundamentally more heterogeneous collection of representations and reasoning techniques to be employed.

7.4 Collecting Narratives From the General Public

The EM-ONE corpus of commonsense narratives is a small hand crafted knowledge base, but for EM-ONE to demonstrate broader competence this knowledge base will need to be grown substantially. I believe one promising approach for doing this is to establish a large-scale effort to collect such narratives from volunteer contributors across the web. I have had success in the past building large scale commonsense knowledge bases by turning to the general public for help. My first such system, Open Mind Common Sense (Singh, 2002b; Singh *et al.*, 2002) attracted an audience of over 15,000 volunteers across the web, who together provided approximately 750,000 items of commonsense

knowledge to a web-based acquisition interface. I have since been involved in the development of several new collection systems that focus more specifically on stories, including the Open Mind Experiences web site (Singh & Barry, 2003), the LifeNet and StoryNet web sites (Singh, Barry, & Liu, 2004) and most recently the ComicKit web site (Williams, Barry, & Singh, 2005). The ComicKit site allows the user to author stories by building comic strips where the characters take actions but can also have internal beliefs and goals expressed in thought bubbles over their heads. While these sites were not designed to accumulate the kinds of narratives specifically used by EM-ONE, it would not be difficult to build a web site where people could tell stories about the physical, social, and mental interactions between Pink and Green.

7.5 Learning Narratives From Experience

In EM-ONE, I chose to encode narratives by hand in order to more directly explore what information a narrative representation should include and how it should be structured. In the future I plan to extend the narrative corpus by learning from problem-solving episodes. I suspect that it is not difficult to learn narratives from more raw records of the experiences of interacting actors, e.g. by recording people acting in the real world (as we and others are beginning to do by instrumenting people and environments with rich arrays of sensors (Pentland *et al.*, 2005)) or in virtual worlds (such as in the popular video game *The Sims*), and generalizing from those experiences. I have also considered setting up a web-based project where hundreds of participants would control the actions of the Roboverse actors to produce scenarios, and where other participants could then annotate those scenarios with aspects of the experiences that are not easily obtained from the raw trace of the actions of the characters, such as the motivations of the actors, what they believe about the situation and about each other, and as well as other aspects of their mental state including their possible deliberations and reflections during the course of the scenario.

7.6 Learning Critics From Experience

It may be possible to learn new critics by analyzing failures in traces of problem-solving episodes. It may be possible, for example, to adapt Soar's chunking learning mechanism

(Laird, Rosenbloom, & Newell, 1986). In Soar's chunking mechanism, the net result of successful problem-solving episodes produce new rules by computing which of the rules applied during the episode contributed directly to producing the successful result, and then "chunking" these applied rules into a single new rule. A very similar mechanism could be used for learning new critics. To learn critics, one would chunk not by computing the causes of successes, but instead by computing the causes of failures. This raises the question of what states should be considered "failures." Perhaps any proposed hypothesis that ends up rejected or action that ends up suppressed could count as a failure. It may be possible to bring to bear existing critics to aid in the credit assignment processes that are engaged during critic learning, because critics seen declaratively relate situations to problems that may result, and so they can be used to hypothesize the causes of failure in new situations.

7.7 Unifying Critics and Narratives

In principle, there need not be so much of a difference between mental critics and those commonsense narratives in which actors encounter and overcome failures. Critics could be embedded in narratives just as commonsense causal relations are embedded in narratives, by relating events to their negative outcomes and those negative outcomes to actions that undo them. Presently, critics are (mostly) procedural and narratives are declarative, but if the frames used by commonsense narratives had attached procedures that corresponded to the operations that mental critics performed, then it would be possible to interpret commonsense narratives so that they had the same procedural behavior as mental critics. The advantage of performing this unification is that mental critics could then deliberate about the content of other mental critics in the same way that they presently deliberate about hypothetical narratives. In EM-TWO, suitably structured "mental critic" commonsense narratives might play some of the roles that procedural mental critics play in EM-ONE.

7.8 Meta-Management by Analogy to Prior Episodes of Thinking

In EM-ONE, the meta-managerial critics are unlike mental critics in one important respect: they do not draw from the EM-ONE narrative corpus to select their actions. This is because I had in mind a few particular overall styles of thinking for EM-ONE. However, I would like in the future to extend EM-ONE so that meta-management engages narrative cases to decide on styles of thinking. It may be possible to write EM-ONE narratives that capture knowledge about higher order aspects of thinking that could then be used to guide deliberation and reflection. Such narratives would provide very high level advice about how thinking should proceed. Consider the following story:

- I considered picking up the stick (initiates deliberation about consequences of the action)
- I wondered if the other person would want the stick (initiates deliberation about the other actor's goals)
- I realized the other person would not want the stick (initiates deliberation with an expected conclusion about the other actor's goals)

This story suggests a particular chain of deliberation that the thinker might engage in. In other words, the reaction-deliberation-reflection cycle is not the only way to structure mental activity—this is just one "story" of how to think. The use of metacritics by themselves gives us great flexibility in structuring cognitive computations, but if metacritics drew from narratives then it would perhaps be simpler to support multiple styles of thinking, because we could author those styles of thinking by authoring narratives. In addition, these styles of thinking could perhaps be learned by reading stories where there are statements about how the characters think about things. Clearly there are limits to the extent to which the processes underlying commonsense thinking can be articulated in a compact narrative, but perhaps even partial and sketchy narratives about thinking could be used to guide the flow of thinking in an AI system.

7.9 Connecting to Cyc

It would not be difficult to implement mental critics on top of the Cyc inference system, which provides inferencing facilities similar to those used by Prolog—in fact, the Cyc inference system is a more general reasoner, capable of a broader class of inferences than ordinary Prolog inference. While I don't expect for the underlying substrate on top of which EM-ONE is built to do very much of the "heavy lifting" of commonsense reasoning, it would be helpful if the substrate allows for more sophisticated queries such as whether one event occurred sometime after another (despite intervening events) or whether a given object is an instance of some general class of object. Also, to extend EM-ONE to a broader range of domains requires expanding its representation, and the most expressive formal representation presently available is the Cyc upper level ontology. Because the current EM-ONE representation is based on frames, it is straightforward to connect to the Cyc system, which contains a wide variety of frame types including frames for many common English words.

7.10 Using Vague and Ambiguous Knowledge

I am interested in the possibility of using *ambiguous* representations during reasoning. Normally one regards ambiguity as a property that should be eliminated from a knowledge representation. But when it comes to knowledge engineering, I see the desire to eliminate all ambiguity in the knowledge representation as the main reason for the so-called *knowledge acquisition bottleneck.* There is an inherent trade-off between ease of acquisition and the precision and accuracy of the collected knowledge. If we could find a way to work with ambiguous knowledge then we would be able to provide knowledge to our systems much more quickly and easily. In addition, some forms of symbol ambiguity can be advantageous if the different senses of a symbol have some useful overlap.[16] In

[16] Generally, I am dubious of the proliferation of symbol names that seems to be the product of most ontological approaches to AI. The problem is that it is difficult to decide when to stop. Pat Hayes once recounted to me a story about a group of ontological engineers arguing about whether a painting that was hanging on the wall was "in" the room—but then an even fiercer argument broke out about whether the paint on the wall was "in" the room! It's certainly an interesting exercise to produce and refine such distinctions, but my sense is that this is not a well-defined task outside of some purpose or goal that guides the production of these distinctions. Stories, however, open up an interesting new possibility. Rather than

EM-TWO, "ambiguity critics" could identify types of ambiguities in narratives and hypotheses, and generate proposals for disambiguated variants of those structures. Each narrative and hypothesis would produce multiple more precise interpretations, and reasoning could proceed in parallel over each of the interpretations. When mistakes are made as a result of an improper disambiguation, reflective critics can then attempt to debug those ambiguity critics.

7.11 Multiple Representations via Panalogy

If one were to compare this thesis with my thesis proposal (Singh, 2002a), perhaps the most glaring omission in EM-ONE, compared to what I proposed to build, are the "panalogy" mechanisms for representing knowledge in multiple ways simultaneously, so as to be able to easily switch between those representations. The term "panalogy" derives from "parallel arrays of analogous representations." In EM-TWO, I would like to include panalogy mechanisms that connect critics and narratives whose contents are partly similar or analogous but use different representation schemes.

In my thesis proposal I had suggested that whenever critics update a representation, which in EM-ONE is done by modifying hypothetical narratives, they actually update multiple representations in parallel. This enables the architecture to quickly switch between different representations because, instead of starting all over each time a new representation is needed, alternate representations are already prepared and ready to go, and suitable critics and narratives will be ready to take over when the present ones run into trouble. One idea that I did not discuss in my thesis proposal was the possibility of using reflective critics to identify situations where such a change in representation was necessary.

---

using a large collection of special symbol names that distinguish between an ever-increasing collection of cases of "in-ness", we can instead say "in" as in the story STORY-532. At some point symbolic distinctions could begin to be made by exploiting their extrinsic contexts of use to provide a basis for further refinement, as opposed to trying to shoehorn that context into the symbol name itself.

Panalogy does not operate by employing any single principle for reformulation, but instead by using a family of techniques that support the synchronizing and sharing of information between different methods concerned with the same or similar problems. By actively maintaining correspondences between multiple representations, we can rapidly switch from one representation to another as work on problems progress. Here are some of the types of parallel representation that I have been considering:

Event panalogies allow maintaining the correspondences between the elements of action and event descriptions across multiple representations. For example, when we imagine the consequences of buying a fancy new car, we can rapidly switch between considering the effects of that purchase on our social status (which it may improve) and on our financial situation (which it may hurt.) This form of panalogy lets us assess the consequences of an action or event from a great many different perspectives at once—for in the ordinary, common sense world, actions and events usually have a wide range of important physical, social, psychological, economic, and other types of consequences. In terms of EM-ONE, the elements of different narratives could be connected so that when one narrative fails to suggest a solution, the knowledge of analogous ones could quickly be brought to bear.

Model panalogies allow maintaining descriptions of different models or interpretations of a situation, like seeing a window simultaneously as both an obstacle and as a portal. Each of these interpretations may suggest different inferences or courses of actions, and if we discover that in fact the window is not locked, inferences based on the "portal" interpretation are already available for use. This form of panalogy is valuable because it takes advantage of the notion that a problem often becomes trivial when we look at it from just the right perspective. A planning problem represented one way might require an immense amount of search, but when represented in another way might be solved by simple hill climbing. Each of these interpretations may suggest different inferences or courses of actions. EM-ONE does not engage in sophisticated forms of classification, but generally, whenever a

situation, object, or event could conceivably be classified in several ways, it may be worth pursuing all of those interpretations in parallel.

Theory panalogies allow maintaining mappings between different theories of the same domain. For example, we may choose to use one theory of time where events are treated as atomic points on a timeline, or use another theory of time where events are treated as occurring over intervals on a timeline. When the first theory is unable to answer a question about, for example, the total duration of some set of actions or the order in which they occurred, we might switch to the second theory. This form of panalogy is useful because it is difficult to find the "best" way to represent fundamental commonsense subjects such as space, time, causality, goals, and so forth. We argue instead that there is no best "upper level ontology" for describing such entities, and that we should instead employ multiple theories about foundational matters. This may require translation tables that allow descriptions in one representation to be translated into the other. This is similar the notion of contexts as described in Guha (1991), which uses "lifting rules" that make explicit the assumptions to add and remove from assertions when transferred it from one context to another. While EM-ONE does not make use of logical theories, something like this idea might be applied to collections of narratives rather than sets of axiomatic rules.

Realm panalogy allow maintaining analogies between different "mental realms," large-scale commonsense domains such as the spatial, temporal, and social realms. Lakoff & Johnson (1990) have argued for example that the knowledge and skills we use for reasoning about space and time are also used to help reason about social realms, for in language there are pervasive metaphors that exist between these seemingly very different domains. This form of panalogy is important because it is clear from language that it is possible to exploit such metaphors to simplify communication about abstract matters, and we suspect that such metaphors may serve similar roles within the mind as well (see Boroditsky (2000) for some recent evidence that temporal ideas have their roots in spatial notions.) The narratives in EM-ONE

combine elements of different realms, but it may be useful, in scaling up the EM-ONE narrative corpus, to begin grouping narratives into more realm-specific stories (e.g. primarily social stories versus primarily physical stories) and then look for systematic ways to connect them.

Abstraction panalogy allow maintaining connections between different abstract descriptions. For example, one might approximate a human skeleton with just a dozen limbs rather than the actual 206 bones of a normal adult, or focusing on particular sub-skeletal structures such as the bones of the right leg. Each of these different abstractions can be linked by their common parts to together form a more realistic or complete model than any individual abstraction could form. This form of panalogy is powerful because it lets us link together a variety of 'simplifications' of a situation, each useful for a different type of problem. If we are trying to grasp a pair of scissors it may be useful to think about each of our fingers separately, but if we are trying to push closed a heavy door we may instead think of the palm of our hand and its five fingers as a single unit that applies pressure to the door. This is related to Minsky's "frame systems" idea (Minsky, 1975). In EM-ONE, one of the difficulties was that particular critics would only match particular structural forms of situation descriptions, and so by using structure panalogies matching could support a variety of different structures for expressing the same idea.

Ambiguity panalogy allow maintaining links between ambiguous senses of predicates. For example, the preposition "in" can refer to a wide range of relations far more specific than any division provided by ordinary dictionary senses. Rather than selecting any particular such relation when describing a situation, we can instead maintain the ambiguity between those relations, which then lets us draw on our understanding of all those related senses to answer questions about how one thing could be "in" another. This form of panalogy lets us bypass one of the basic difficulties in building symbolic systems—namely, that it is incredibly challenging and perhaps impossible to define any given symbol precisely enough that we and others will use it only as intended in the future. Just as the meanings of words evolve

with their use, and quickly come to acquire multiple new senses in different contexts, so should the meanings of symbols. In EM-TWO, I hope to incorporate ambiguity panalogies to simplify the engineering of intricate stories. In Cyc, one is forced to provide far more detail than is convenient, or even practically possible. E.g. one is usually forced to pick out the most specific "in" relation that is meant in a given situation, even when in principle the specific sense could have been determined by context or other types of inference.

7.12 Structural Critics

Building large commonsense knowledge bases is a difficult, error-prone process. In (Singh, 2003d), I propose the use of *structural critics* that recognize syntax errors and form errors of various types within a commonsense knowledge base, e.g. misspellings, mistaken terms, unnecessary vagueness and ambiguity, incorrect role or slot assignments, and so forth. I proposed these critics as a way to deal with the variety of errors that existed in the Open Mind Common Sense (OMCS) knowledge base, a corpus of commonsense facts that I collected from the general public over the web (Singh, 2002b; Singh *et al.*, 2002). Because the general public is untrained in knowledge engineering methods, the knowledge is often not

- at the right level of detail,
- suitably contextualized,
- completely expressed,
- expressed in a uniform enough vocabulary,
- sufficiently unambiguous,

and it suffers from other such "structural" problems that lead to difficulties during reasoning. I sketched a preliminary collection of structural critics that could recognize a few of the kinds of errors that showed up within the OMCS corpus. Structural critics notice problems with the syntactic form in which knowledge is expressed, as opposed to problems with the content of the knowledge itself:

- **Missing context.** A given item of knowledge has implicit contextual assumptions that could be made explicit.
- **Incomplete variable bindings.** A given item of knowledge does not fully specify all of the important individuals involved.
- **Too general.** A given item of knowledge seems to be making a very broad generalization.
- **Varying expressions.** A concept or relation is expressed in different ways in different parts of the knowledgebase.
- **Symbol used ambiguously.** A given symbol is used ambiguously in different parts of the knowledgebase.

These are intended as a preliminary step towards a richer classification of the kinds of defects that knowledge bases could contain. In that paper, I focused on commonsense facts expressed in English, but every type of knowledge representation has the potential for different types of bugs, and so we might need to accumulate different structural critics for different representations.

If EM-ONE possessed a collection of such structural critics, perhaps it could deal with knowledge that is expressed less carefully. Such knowledge may be easier to gather quickly, such as commonsense narratives extracted from reading natural language descriptions of the experience of solving problems, contributed by people, or extracted from books or the text of the web.

7.13 The Dynamics of Critic Systems

In the long run, as we move towards critic systems with very large numbers of critics, and where at any time a subset of critics are active and the others quiescent, I would like to move to a model where the top-level goal of metacritics to be to reduce the number of critics that could potentially apply. This may seem contradictory, since metacritics act to increase the number of active critics by selecting other critics. However, with more active critics, it is more likely that problems will get solved, which reduces the number of critics. Thus thinking can be regarded as a battle between identifying problems in the

world and the mind, thus spawning new critics, and solving those problems by taking actions in the world and the mind, which quiets active critics.

What would be the dynamics of such complex societies of critics? I expect there will be interesting and difficult types of instabilities, such as manic, confusing explosions where too many critics fire, or depressive, ineffective quiescence where too few critics fire. Perhaps critics could meander off course, spending all of their time on subproblems that are not really relevant to any higher-level goal. I suspect that, to scale up the EM-ONE architecture, we will need an entirely new set of critics that are primarily "regularity," concerned with preventing wild oscillations or "mood swings" in the sets of critics that are active at any moment.[17]

7.14 Probabilistic Inference

Statistical representations, despite their present day popularity, are not used in EM-ONE. This was primarily because I did not have a clear method in mind for representing commonsense narrative structures probabilistically and for providing an underlying reasoning substrate for probabilistic representations that supported the types of basic reasoning tasks that Prolog provides for rule-based representations. That said, I believe it is worth exploring how to build EM-ONE on top of a probabilistic substrate. There are several ways one might proceed:

- Narratives could allow some uncertainty in their structure. One way to do this is to allow for frame slots to point not to particular other frames, but instead a probability

[17] In humans we might call these kinds of bugs mood disorders. Here is a naive theory of several human mood disorders: in bipolar disorder one switches uncontrollably between purely critics and purely advocates; in unipolar depression one's critics take control; in cyclothymia one switches between their weaker critics and weaker advocates; and in euphoria one's advocates take control (American Psychiatric Association, 2000). Perhaps each of these disorders is due to some broken reflective critic whose job is to notice and prevent these particular types of bugs in the large-scale activity of mental critics. While chemical treatments (such as lithium) are fairly effective in treating these types of psychopathologies, it seems to me that such treatments work largely by damping the mood swings. We need to better understand the etiology of these problems to provide more effective treatments.

distribution of other frames. This is what is done, for example, in the P-CLASSIC probabilistic frame system (Koller & Pfeffer, 1998).

- The process of retrieving and matching narratives could be done using some form of probabilistic inference. Given a context, some sort of probabilistic subsumption calculation might be performed to identify those narratives that seem to match best the current context.

- Critics could act not deterministically but with some probability. Thinking would then be a more stochastic process. This could be done by having metacritics not activate particular sets of critics with absolute certainty, but instead control the parameters of a distribution over their probable activation.

The difficulty that is often not noted with such methods is that probabilistic methods are computationally often worse even than methods based using logical models. Exact computation of probabilistic inference in general belief networks is NP-hard (Cooper, 1990). It turns out that even approximating probabilistic inference in general belief network is NP-hard (Dagum & Luby, 1993; Roth, 1996).

To me this means that an architecture such as EM-ONE, one that helps organize a variety of heuristic methods, is even more desperately needed for probabilistic representations than it is for logical representations. This is especially the case for commonsense reasoning, where we must cope with potentially millions of units of knowledge and, given an ontology the size of the Cyc ontology, the virtually unlimited number of potential hypothetical scenarios that can be expressed.

7.15 Engaging EM-ONE for Perception and Motor Control

The Roboverse simulator includes support for synthetic vision and realistic torque and velocity based motor control. However, the access that EM-ONE has to the simulator environment is based on small number of simple perceptual predicates and behavioral routines. Even in the simulated Roboverse environment, there are many details about the

way the world works that are critically dependent on the fine details of the shape of space, and one's access to those details is mediated by first person visual, auditory, and haptic representations. Even just modeling the many potential arrangements of sticks and boards, and the many potential kinematic arrangements of the robot in connection with those arrangements, requires a richer representation than the one used by EM-ONE, one that takes into account the finer geometric configurations involved. I am looking forward to a future version of the architecture that engages substantial commonsense spatial, physical, and bodily knowledge to the problem of perception and motor control itself, involving more specialized visual and haptic representations.

7.16 Evaluation Methods

In recent years, AI research has moved towards problems that lend themselves to easy evaluation. I believe that one of the reasons machine learning is so popular today is because there is a clear and simple way to evaluate most machine learning systems. Can we develop methodologies for evaluating commonsense reasoning systems? This is challenging because, unlike many other types of reasoning, it is not a single type of skill. Instead, commonsense reasoning involves a heterogeneous array of abilities. Because commonsense thinking involves a great variety of skills, evaluation is a challenging issue, but I consider it a high priority for future work. In the case of EM-ONE, there are several possible approaches we might pursue to evaluate the performance of the architecture:

Evaluating mental critics independently. As we accumulate a finer catalog of types of mental critics, it may be possible to evaluate each of them independently. For example, one type of mental critic may have the task of predicting what the next situation might be, given an initial situation. It may be possible to compare the performance of EM-ONE where different such mental critics are swapped in for the same task.

Comparing truncated version of the architecture. It may be possible to evaluate the value of additional reflection. One might compare the performance of EM-ONE on some problem with a truncated version of the architecture where its higher levels have been ablated, e.g. where it only reacts but does not deliberate or reflect, or where it reacts and

deliberates but does not reflect. I would expect that EM-ONE would perform reasonably with just a reactive layer, but by adding some deliberation its performance should improve, and adding a reflective layer should lead to further improvements in the long run.

Evaluating against a catalog of commonsense scenarios. Because common sense involves so many different types of knowledge and reasoning skills, no simple, uniform metric can measure one's ability at commonsense reasoning. So rather than trying to maximize some simple score, like what percentage of the words of a document are part-of-speech tagged correctly, I propose that we endorse a more heterogeneous approach—we should accumulate a large catalog of commonsense scenarios, and assess the performance of our systems by how many of those scenarios they are capable of emulating, perhaps taking into account factors like how well a system adapts to mild perturbations to those scenarios. To simplify development, these scenarios should at first not depend on advanced forms of adult thinking, and instead they should be kinds of scenarios that one might expect even a young child to be capable of emulating. In (Minsky, Singh, & Sloman, 2004) we propose an example of a series of scenarios of increasing difficulty:

1. Person wants to get box from high shelf. Ladder is in place. Person climbs ladder, picks up box, and climbs down.

2. As for 1, except that the person climbs ladder, finds he can't reach the box because it's too far to one side, so he climbs down, moves the ladder sideways, then as 1.

3. As for 1, except that the ladder is lying on the floor at the far end of the room. He drags it across the room lifts it against the wall, then as 1.

4. As for 1, except that if asked while climbing the ladder why he is climbing it the person answers: something like "To get the box." It should understand why "*To*

get to the top of the ladder" or "*To increase my height above the floor*" would be inappropriate, albeit correct.

5. As for 2 and 3, except that when asked, "Why are you moving the ladder?" the person gives a sensible reply. This can depend in complex ways on the previous contexts, as when there is already a ladder closer to the box, but which looks unsafe or has just been painted. If asked, "would it be safe to climb if the foot of the ladder is right up against the wall?" the person can reply with an answer that shows an understanding of the physics and geometry of the situation.

6. The ladder is not long enough to reach the shelf if put against the wall at a safe angle for climbing. Another person suggests moving the bottom closer to the wall, and offers to hold the bottom of the ladder to make it safe. If asked why holding it will make it safe, gives a sensible answer about preventing rotation of ladder.

7. There is no ladder, but there are wooden rungs, and rails with holes from which a ladder can be constructed. The person makes a ladder and then acts as in previous scenarios. (This needs further unpacking, e.g. regarding sensible sequences of actions, things that can go wrong during the construction, and how to recover from them, etc.)

8. As for 7, but the rungs fit only loosely into the holes in the rails. Person assembles the ladder but refuses to climb up it, and if asked why can explain why it is unsafe.

9. Person watching another who is about to climb up the ladder with loose rungs should be able to explain that a calamity could result, that the other might be hurt, and that people don't like being hurt.

It is not difficult to come up with more such commonsense scenarios. The commonsense reasoning community has already recognized the value of this approach to evaluating and

comparing approaches, e.g. Leora Morgenstern is maintaining a web site that contains a broad range of "commonsense problem types" that any commonsense system should be able to cope with and that serve as benchmarks and challenges for the commonsense reasoning community (Morgenstern, 2002). If these scenarios were all defined as challenges within the Roboverse microworld, then perhaps solutions to these problems would have a greater chance of being combined into a single integrated system.

Chapter 8

Contributions

This thesis made several contributions:

- A commonsense reasoning system that can debug its own reasoning processes when it makes mistakes. The reflective reasoning framework presented here could lead to systems that are more tolerant to imperfect components, both in their reasoning processes and in their knowledge.

- A critic language with which one can author ways to deliberate, as well as ways to criticize and repair the processes underlying that deliberation, and examples of "programs" (critic networks) written in this language. This language could lead to a new generation of AI systems organized as arrays of heuristic methods for solving multiple problem-types both in the world and in the systems themselves.

- The beginnings of a classification of the types of criticisms that one might make of narrative hypotheses, and a classification of the types of reasoning failures that can occur in commonsense thinking. This classification could lead to the promotion of mistakes to first-class objects within AI, as entities that are explicitly represented and studied.

- An example of how to build an AI system that combines reactive, deliberative, and reflective processes across the physical, social, and mental commonsense realms. This architectural design could lead to a new generation of commonsense AI systems, especially commonsense-enabled robots, that not only act in the physical world, but also have rich social and mental lives.

- A first implementation of Marvin Minsky's Emotion Machine model of commonsense intelligence. When Minsky's book *The Society of Mind* was published, implementations of the theory did not follow shortly thereafter. I hope that this thesis leads to many more implementations of Minsky's ideas, by giving workers in AI one example of how the theories in *The Emotion Machine* can be brought to life.

References

American Psychiatric Association (Task Force on DSM-IV). (2000). *Diagnostic and statistical manual of mental disorders: DSM-IV-TR*. Washington, DC: American Psychiatric Association.

Beaudoin, L. P. (1994). Goal processing in autonomous agents. PhD thesis. School of Computer Science, The University of Birmingham, England.

Boroditsky, L. (2000). Metaphoric structuring: understanding time through spatial metaphors. *Cognition*, *75*(1), 1-28.

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

Brooks, R. (1990). The Behavior Language: User's Guide. MIT AI Lab Memo 1227.

Cohen, P., & Levesque, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42, 213-261.

Cooper, F. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42, 393-405.

Cox, M. (1997). An explicit representation of reasoning failures. In D. B. Leake & E. Plaza (Eds.), Case-Based Reasoning Research and Development: Second International Conference on Case-Based Reasoning (pp. 211-222). Berlin: Springer-Verlag.

Cox, M., & Ram, A. (1999). Introspective multistrategy learning: On the construction of learning strategies. *Artificial Intelligence*, 112, 1-55.

Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60, 141-153.

Davis, E., & Morgenstern, L. (2004). Introduction: progress in formal commonsense reasoning. *AI Journal: Special Issue on Logical Formalizations and Commonsense Reasoning*. *153*(1-2), 1-12.

Fox, S., & Leake, D. (1995). Using introspective reasoning to refine indexing. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. Chambery, France.

Gordon, A., & Hobbs, J. (2004). Formalizations of commonsense psychology. *AI Magazine*, *25*(4), 49-62.

Gordon, A. (2004). The representation of planning strategies. *Artificial Intelligence*, 153, 287-305.

Gordon, A., & Nair, A. (2004). Expressions related to knowledge and belief in children's speech. *Proceedings of the 26th Annual Meeting of the Cognitive Science Society (CogSci-2004)*. Chicago. Mahwah, NJ: Lawrence Erlbaum Associates.

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

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

Guha, R. V. (1991). *Contexts: A Formalization and Some Applications*. PhD Thesis, Department of Computer Science, Stanford.

James, W. (1890). *The Principles of Psychology*. Harvard University Press, 1983. Reprint of the 1890 edition.

Koller, D., & Pfeffer, A. (1998). Probabilistic frame-based systems. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. Madison, Wisconsin.

Kolodner, J. (1993). *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann.

Laird, J., & Rosenbloom, P.S. (1996). The evolution of the Soar cognitive architecture. In D. M. Steier & T. M. Mitchell, (Eds.), *Mind Matters: A tribute to Allen Newell*, pp. 1-50, Mahwah, NJ: Erlbaum.

Laird, J., Rosenbloom, P., & Newell, A. (1986). Chunking in Soar: the anatomy of a general learning mechanism. *Machine Learning*, *1*(1), 11-46.

Lakoff, G., & Johnson, M. (1990). *Metaphors We Live By*. Chicago, IL: University of Chicago Press.

Lenat, D. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*(11), 33-38.

McCarthy, J. (1995). Making robots conscious of their mental states. *Machine Intelligence*, 15, 3-17.

McCarthy, J., Minsky, M., Sloman, A., Gong, L., Lau, T., Morgenstern, L., Mueller, E. T., Riecken, D., Singh, M., & Singh, P. (2002). An architecture of diversity for commonsense reasoning. *IBM Systems Journal*, *41*(3), 530-539.

Minsky, M. (1965). Mind, matter and models. *Proceedings of the International Federation of Information Processing Congress*, 1, 45-49. Cambridge, MA: MIT Press.

Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), *The Psychology of Computer Vision*, pp. 211-277. New York, NY: McGraw-Hill.

Minsky, M. (1986). *The Society of Mind.* New York: Simon and Schuster.

Minsky, M. (1992). Future of AI technology. *Toshiba Review*, *47*(7).

Minsky, M. (forthcoming). *The Emotion Machine.*

Minsky, M., & Papert, S. (1972). "Progress Report on Artificial Intelligence" AI Memo 252, MIT Artificial Intelligence Laboratory, Cambridge, MA, Jan 1972.

Minsky, M., Singh, P., & Sloman, A. (2004). The St. Thomas common sense symposium: designing architectures for human-level intelligence, *25*(2), 113-124.

Morgenstern, L. (2002). Common Sense Problem Page. Retrieved 13 May 2005 from: http://www-formal.stanford.edu/leora/cs/

Mueller, E. T. (1998). *Natural Language Processing with ThoughtTreasure*. New York: Signiform.

Mueller, E. T. (1999). *Prospects for in-depth story understanding by computer*. CogPrints cog00000554.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA.: Harvard University Press.

Newell, A., Shaw, J., &, Simon., H. (1960a). Report on a general problem solving program. *Proceedings of the International Conference on Information Processing*. Paris: UNESCO, pp. 256-64.

Newell, A., Shaw, J., & Simon, H. (1960b). A variety of intelligent learning in a General Problem Solver, in Yovits M.C., & Cameron, S. (Eds.), *Self-Organizing Systems*, pp. 153-189, Elmford, NY: Pergammon Press.

Papert, S. (2004). The turtle's long slow trip: macro-educological perspectives on microworlds. Retrieved 13 May 2005 from: http://www.iaete.org/soapbox/microworlds.cfm

Pentland, A., Choudhury, T., Eagle, N., & Singh, P. (2005). Human Dynamics: Computation for Organizations. *Pattern Recognition Letters*, 26, 503-511.

Rao, A., & Georgeff, M. (1991). Deliberation and intentions. *Proceedings of 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles. Los Angeles, CA.

Rao, A., & Georgeff, M. (1995). BDI Agents: From Theory to Practice. *Proceedings of the First International Conference on Multiagent Systems*. San Francisco, CA.

Schank, R. & Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Earlbaum Assoc.

Schank, R. (1982). *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press.

Schank, R. (1986). *Explanation Patterns: Understanding Mechanically and Creatively.* Hillsdale, NJ: Erlbaum.

Singh, P. (2002a). The Panalogy Architecture for Commonsense Computing. PhD Thesis Proposal. Department of Electrical Engineering and Computer Science, MIT.

Singh, P. (2002b). The public acquisition of commonsense knowledge. *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access.* Palo Alto, CA.

Singh, P. (2003a). The Panalogy architecture for commonsense computing - Brief Description. Report for the Institute for Defense Analysis. Unpublished report.

Singh, P. (2003b). A preliminary collection of reflective critics for layered agent architectures. *Proceedings of the Safe Agents Workshop (AAMAS 2003).* Melbourne, Australia.

Singh, P. (2003c). Reaching for dexterous manipulation (Area Exam). Department of Electrical Engineering and Computer Science, MIT.

Singh, P. (2003d). Structural critics for commonsense knowledge bases. Unpublished manuscript. Retrieved 13 May 2005 from: http://web.media.mit.edu/~push/StructuralCritics.html

Singh, P., & Barry, B. (2003). Collecting commonsense experiences. *Proceedings of the Second International Conference on Knowledge Capture (K-CAP 2003).* Sanibel Island, FL.

Singh, P., & Minsky, M. (2003). An architecture for combining ways to think. *Proceedings of the International Conference on Knowledge Intensive Multi-Agent Systems.* Cambridge, MA.

Singh, P., & Minsky, M. (2005). An architecture for cognitive diversity. *Visions of Mind,* Darryl Davis (Ed.), London: Idea Group, Inc.

Singh, P., & Williams, W. (2003). LifeNet: a propositional model of ordinary human activity. *Proceedings of the Workshop on Distributed and Collaborative Knowledge Capture (DC-KCAP) at K-CAP 2003.* Sanibel Island, FL.

Singh, P., Barry, B., & Liu, H. (2004). Teaching machines about everyday life. *BT Technology Journal, 22*(4), 227-240.

Singh, P., Minsky, M., & Eslick, I. (2004). An architecture for computing common sense. *BT Technology Journal, 22*(4), 201-210.

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., & Zhu, W. L. (2002). Open Mind Common Sense: Knowledge acquisition from the general public. *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. Lecture Notes in Computer Science.* Heidelberg: Springer-Verlag.

Sloman, A. (2001). Beyond Shallow Models of Emotion. *Cognitive Processing, 2*(1), 178-198.

Soar Technology (2002). Soar: a functional approach to general intelligence. Retrieved 13 May 2005 from: http://www.eecs.umich.edu/~soar/docs/SoarFunctionalOverview.pdf

Sussman, G. J. (1973). A computational model of skill acquisition. PhD thesis. Department of Mathematics, MIT.

Turing, A. (1950). Computing machinery and intelligence. *Mind, 59*(236), 433-460.

Williams, R., Barry, B., & Singh, P. (2005). ComicKit: acquiring story scripts using commonsense feedback. *Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI 2005).* San Diego, CA.

Appendix A

Expanded Form of Mental Critic

The **defcritic** operator is implemented as a Common Lisp macro that expands the given critic expression to a more complicated Common Lisp expression that involves a substantial call to the Prolog subsystem. For example, the following very simple critic

```
(defcritic (reactive*actor-intends-action=>do-action)
  (in conditions current-conditions
    (intends :actor ACTOR :prop (ACTION :actor ACTOR :object OBJECT)))
  (=>)
  (in conditions current-conditions
    (assert (does :actor ACTOR :prop (ACTION :actor ACTOR :object OBJECT)))))
```

expands to expression shown below. Most critics are more complicated than this one.

```
(progn (in-package :think)
  (<-- (reactive*actor-intends-action=>do-action)
    (lisp ?CRITICISM (intern (gensym "T")))
    (or (and (lispp (not (null *CRITIC-STACK*)))
             (lisp ?PARENT (caar *CRITIC-STACK*))
             (assert ((called-by ?CRITICISM ?PARENT))))
        (true))
    (!)
    (lisp (when *VERBOSE*
            (format t "~S~%" (list 'reactive*actor-intends-action=>do-action))))
    (lisp (when *TRACING*
       (if (not (null *CRITIC-STACK*))
           (setf (aref *REFLECTIVE-TRACE* *REFLECTIVE-TRACE-POINTER*)
                 (list 'calls (caar *CRITIC-STACK*)
                       (cons ?CRITICISM
                             (list 'reactive*actor-intends-action=>do-action))))
         (setf (aref *REFLECTIVE-TRACE* *REFLECTIVE-TRACE-POINTER*)
               (list 'calls nil
                     (cons ?CRITICISM
                           (list 'reactive*actor-intends-action=>do-action)))))
       (incf *REFLECTIVE-TRACE-POINTER*)))
    (lisp (setf *CRITIC-STACK*
                (cons (cons ?CRITICISM
                            (list 'reactive*actor-intends-action=>do-action))
                      *CRITIC-STACK*)))
    (holds conditions current-conditions (subsit current-conditions ?SIT_43303))
```

```
  (holds conditions current-conditions (truth ?SIT_43303 true))
  (holds conditions current-conditions (isa ?SIT_43303 situation))
  (holds conditions current-conditions (type ?SIT_43303 intends))
  (holds conditions current-conditions (actor ?SIT_43303 ?ACTOR))
  (holds conditions current-conditions (truth ?SIT_43304 true))
  (holds conditions current-conditions (isa ?SIT_43304 situation))
  (holds conditions current-conditions (type ?SIT_43304 ?ACTION))
  (holds conditions current-conditions (subsit ?SIT_43303 ?SIT_43304))
  (holds conditions current-conditions (actor ?SIT_43304 ?ACTOR))
  (holds conditions current-conditions (object ?SIT_43304 ?OBJECT))
  (holds conditions current-conditions (prop ?SIT_43303 ?SIT_43304))
  (lisp ?N_SIT_43308_ (intern (gensym "N-N_SIT_43308_")))
  (lisp ?N_SIT_43307_ (intern (gensym "N-N_SIT_43307_")))
  (lisp ?N_TOPSIT_43305_ (intern (gensym "N-N_TOPSIT_43305_")))
  (lisp ?FACT43321 (list 'subsit ?N_TOPSIT_43305_ ?N_SIT_43307_))
  (assert-no-dups ((item conditions current-conditions ?FACT43321 F-43322)))
  (lisp ?FACT43323 (list 'truth ?N_SIT_43307_ 'true))
  (assert-no-dups ((item conditions current-conditions ?FACT43323 F-43324)))
  (lisp ?FACT43325 (list 'isa ?N_SIT_43307_ 'situation))
  (assert-no-dups ((item conditions current-conditions ?FACT43325 F-43326)))
  (lisp ?FACT43327 (list 'type ?N_SIT_43307_ 'does))
  (assert-no-dups ((item conditions current-conditions ?FACT43327 F-43328)))
  (lisp ?FACT43329 (list 'actor ?N_SIT_43307_ ?ACTOR))
  (assert-no-dups ((item conditions current-conditions ?FACT43329 F-43330)))
  (lisp ?FACT43331 (list 'truth ?N_SIT_43308_ 'true))
  (assert-no-dups ((item conditions current-conditions ?FACT43331 F-43332)))
  (lisp ?FACT43333 (list 'isa ?N_SIT_43308_ 'situation))
  (assert-no-dups ((item conditions current-conditions ?FACT43333 F-43334)))
  (lisp ?FACT43335 (list 'type ?N_SIT_43308_ ?ACTION))
  (assert-no-dups ((item conditions current-conditions ?FACT43335 F-43336)))
  (lisp ?FACT43337 (list 'subsit ?N_SIT_43307_ ?N_SIT_43308_))
  (assert-no-dups ((item conditions current-conditions ?FACT43337 F-43338)))
  (lisp ?FACT43339 (list 'actor ?N_SIT_43308_ ?ACTOR))
  (assert-no-dups ((item conditions current-conditions ?FACT43339 F-43340)))
  (lisp ?FACT43341 (list 'object ?N_SIT_43308_ ?OBJECT))
  (assert-no-dups ((item conditions current-conditions ?FACT43341 F-43342)))
  (lisp ?FACT43343 (list 'prop ?N_SIT_43307_ ?N_SIT_43308_))
  (assert-no-dups ((item conditions current-conditions ?FACT43343 F-43344)))
  (lisp (setf *CRITIC-STACK* (cdr *CRITIC-STACK*))))
(defun reactive*actor-intends-action=>do-action ()
  (prolog (reactive*actor-intends-action=>do-action)))
(<-- (assess-reactive*actor-intends-action=>do-action)
  (lisp ?CRITICISM (intern (gensym "T")))
  (or (and (lispp (not (null *CRITIC-STACK*)))
           (lisp ?PARENT (caar *CRITIC-STACK*))
           (assert ((called-by ?CRITICISM ?PARENT))))
      (true))
  (!)
  (lisp (when *VERBOSE*
          (format t "~S~%" 'assess-reactive*actor-intends-action=>do-action)))
  (lisp (when *TRACING*
     (if (not (null *CRITIC-STACK*))
         (setf (aref *REFLECTIVE-TRACE* *REFLECTIVE-TRACE-POINTER*)
               (list 'calls (caar *CRITIC-STACK*)
                     (cons ?CRITICISM
```

```
                          (list 'assess-reactive*actor-intends-action=>do-action))))
      (setf (aref *REFLECTIVE-TRACE* *REFLECTIVE-TRACE-POINTER*)
            (list 'calls nil
                  (cons ?CRITICISM
                        (list 'assess-reactive*actor-intends-action=>do-action)))))
    (incf *REFLECTIVE-TRACE-POINTER*)))
  (lisp (setf *CRITIC-STACK*
              (cons (cons ?CRITICISM
                          (list 'reactive*actor-intends-action=>do-action))
                    *CRITIC-STACK*)))
  (holds conditions current-conditions (subsit current-conditions ?SIT_43347))
  (holds conditions current-conditions (truth ?SIT_43347 true))
  (holds conditions current-conditions (isa ?SIT_43347 situation))
  (holds conditions current-conditions (type ?SIT_43347 intends))
  (holds conditions current-conditions (actor ?SIT_43347 ?ACTOR))
  (holds conditions current-conditions (truth ?SIT_43348 true))
  (holds conditions current-conditions (isa ?SIT_43348 situation))
  (holds conditions current-conditions (type ?SIT_43348 ?ACTION))
  (holds conditions current-conditions (subsit ?SIT_43347 ?SIT_43348))
  (holds conditions current-conditions (actor ?SIT_43348 ?ACTOR))
  (holds conditions current-conditions (object ?SIT_43348 ?OBJECT))
  (holds conditions current-conditions (prop ?SIT_43347 ?SIT_43348))
  (lisp (setf *CRITIC-STACK* (cdr *CRITIC-STACK*)))
  (lisp ?hyp ?H)
  (assert ((criticism assess-reactive*actor-intends-action=>do-action ?hyp))))
(defun assess-reactive*actor-intends-action=>do-action ()
  (prolog (assess-reactive*actor-intends-action=>do-action)))
```

Appendix B

Expanded Form of Commonsense Narrative

The **defnarrative** operator is implemented as a Common Lisp macro that expands the given narrative expression to a more complicated Common Lisp expression that involves a substantial call to the Prolog subsystem. For example, the following narrative

```
(defnarrative does-not-observe-actor-intent
  (desires green (is-attached stick board))
  (sequential
    (does green (attaches green stick board))
    (not (observes pink (does green (attaches green stick board)) [1]))
    (believes pink (not (desires green (is-attached stick board))) [2]))
  (causes [1] [2]))
```

expands to expression shown below.

```
(prolog
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (subsit does-not-observe-actor-intent SIT_43383) F-43395)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (truth SIT_43383 true) F-43396)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (isa SIT_43383 situation) F-43397)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (type SIT_43383 desires) F-43398)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (actor SIT_43383 green) F-43399)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (truth SIT_43384 true) F-43400)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (isa SIT_43384 situation) F-43401)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (type SIT_43384 is-attached) F-43402)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (subsit SIT_43383 SIT_43384) F-43403)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (subject SIT_43384 stick) F-43404)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (object SIT_43384 board) F-43405)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
                         (prop SIT_43383 SIT_43384) F-43406)))
  (assert-no-dups ((item narratives does-not-observe-actor-intent
```

```
                    (isa SIT_43385 situation) F-43407)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (subsit does-not-observe-actor-intent SIT_43385) F-43408)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (truth SIT_43386 true) F-43409)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (isa SIT_43386 situation) F-43410)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (type SIT_43386 does) F-43411)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (subsit SIT_43385 SIT_43386) F-43412)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (actor SIT_43386 green) F-43413)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (truth SIT_43387 true) F-43414)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (isa SIT_43387 situation) F-43415)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (type SIT_43387 attaches) F-43416)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (subsit SIT_43386 SIT_43387) F-43417)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (actor SIT_43387 green) F-43418)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (object SIT_43387 stick) F-43419)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (target SIT_43387 board) F-43420)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (prop SIT_43386 SIT_43387) F-43421)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (truth SIT_43388 false) F-43422)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (isa SIT_43388 situation) F-43423)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (type SIT_43388 observes) F-43424)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (subsit SIT_43385 SIT_43388) F-43425)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (actor SIT_43388 pink) F-43426)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (truth SIT_43389 true) F-43427)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (isa SIT_43389 situation) F-43428)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (type SIT_43389 does) F-43429)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (subsit SIT_43388 SIT_43389) F-43430)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (actor SIT_43389 green) F-43431)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (truth SIT_43390 true) F-43432)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (isa SIT_43390 situation) F-43433)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                    (type SIT_43390 attaches) F-43434)))
```

```
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (subsit SIT_43389 SIT_43390) F-43435)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (actor SIT_43390 green) F-43436)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (object SIT_43390 stick) F-43437)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (target SIT_43390 board) F-43438)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (prop SIT_43389 SIT_43390) F-43439)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (prop SIT_43388 SIT_43389) F-43440)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (follows SIT_43386 SIT_43388) F-43441)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (truth SIT_43391 true) F-43442)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (isa SIT_43391 situation) F-43443)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (type SIT_43391 believes) F-43444)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (subsit SIT_43385 SIT_43391) F-43445)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (actor SIT_43391 pink) F-43446)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (truth SIT_43392 false) F-43447)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (isa SIT_43392 situation) F-43448)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (type SIT_43392 desires) F-43449)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (subsit SIT_43391 SIT_43392) F-43450)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (actor SIT_43392 green) F-43451)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (truth SIT_43393 true) F-43452)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (isa SIT_43393 situation) F-43453)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (type SIT_43393 is-attached) F-43454)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (subsit SIT_43392 SIT_43393) F-43455)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (subject SIT_43393 stick) F-43456)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (object SIT_43393 board) F-43457)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (prop SIT_43392 SIT_43393) F-43458)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (prop SIT_43391 SIT_43392) F-43459)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (follows SIT_43388 SIT_43391) F-43460)))
(assert-no-dups ((item narratives does-not-observe-actor-intent
                       (causes SIT_43388 SIT_43391) F-43461)))))
```