Anti-Boxology:
Agent Design in Cultural Context

Phoebe Sengers

August 1998

CMU-CS-98-151

Computer Science Department and
Program in Literary and Cultural Theory
Carnegie Mellon University
Pittsburgh, PA

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Thesis Committee:

Joseph Bates, chair
Camilla Griggers
Jill Fain Lehman
Simon Penny

This work was supported in part by a National Science Foundation Graduate Research Fellowship, and by the Office of Naval Research under grant N00014-92-J-1298.

Keywords: Believable agents, action-selection, behavior-based AI, behavior integration, postmodernism, schizophrenia, cultural studies of science, critical technical practice

Abstract

Artificial Intelligence (AI), the design of technology with attributes that we traditionally associate with living beings, generally follows the broader scientific tradition of focusing on technical problems and their solutions within a relatively constrained framework. The cultural studies of science, on the other hand, insists that scientific work should be understood as it springs from and influences other cultural phenomena, including the background of metaphors and assumptions that influence the way scientists do their work. In this thesis, I explore the possibilities for AI and the cultural studies of science to engage in a mutually beneficial alliance, by studying AI as a culturally situated activity and by using results of that study to generate novel technology.

Specifically, I focus on the design of *autonomous agents*, programs which are intended to represent a complete person, animal, or character. In the alternative AI tradition, these agents are created from a set of independent building blocks termed *behaviors*. A major open question is how these behaviors can be synthesized to create an agent with overall coherent behavior. I trace the problems in behavior integration to a strategy called *atomization* that AI shares with industrialization and psychiatric institutionalization. Atomization is the process of breaking agents into modular chunks with limited interaction and represents a catch-22 for AI; while this strategy is essential for building understandable code, it is fatal for creating agents that have the overall coherence we have come to associate with living beings.

I tackle this problem of integration by redefining the notion of agent. Instead of seeing agents as autonomous creatures with little reference to their sociocultural context, I suggest that agents can be thought of in the style of cultural studies as a form of communication between the agent's designer and the audience which will try to comprehend the agent's activity. With this metaphor as a basis, it becomes clear that we need to integrate, not the agent's internally defined code, but the way in which the agent presents itself to the user. Narrative psychology suggests that agents will be maximally comprehensible as intentional beings if they are structured to provide cues for narrative. I therefore build an agent architecture, the *Expressivator*, which provides support for narratively comprehensible agents, most notably by using behavioral transitions to link atomic behaviors into narrative sequences.

Acknowledgements

This thesis is dedicated with gratitude to my family, who gave me support, encouragement, and many entertaining hours of sharp-tongued wit during the years it took to do this work. It is especially dedicated to Nishka, my beloved companion throughout my education, who died, sadly, only a few short weeks before it was over.

I have been fortunate at Carnegie Mellon to be surrounded by many lively, intelligent, and stimulating friends and colleagues. First and foremost, I want to thank my advisors, Joe Bates and Camilla Griggers, to whom I am in great debt.

I was lucky to fall into Joe's lap when I came to CMU. He is one of the smartest people I know, but for me his unique gift in a university full of smart people is that he understands and values the connection between work and life. He is a visionary, and as such knows that work should not only go well but *feel* right, and the kinds of things a person can do to find that right work and go about it, happily. For years, Joe has helped me in innumerable ways to follow my dream of synthesizing AI and cultural studies, even when at times the work that involved seemed to him to be incomprehensible or even bizarre. Things would have gone very differently without him, and I am happy, at the end, to look back and see the firm stamp of his thinking in my work.

Camilla has been a fabulous resource, challenging me to strengthen the theoretical aspects of the dissertation while (perhaps more importantly) helping me learn how to do this work without going insane. two of her greatest gifts to me have been guidance in figuring out what *I* want to do and support in figuring out how to do it. She is also the only advisor I know who, when confronted with a stressed-out student, would give them a foot massage. Camilla, thanks for the support of mind, body, and soul!

I am also grateful to my other committee members, Jill Fain Lehman and Simon Penny, who provided me with valuable comments and advice. Jill's views on AI are both sharp as a tack and very different from mine, and as a consequence her feedback — not to mention her patience with a research area that at times must have seemed from another planet — was enormously helpful. This was Simon's first Ph.D. committee, a fact that was made clear by his sheer enthusiasm; he has not yet learned that a committee member's proper function is to try to avoid interaction with his or her student until the last possible moment. Many thanks to you, Simon, for the frequent meetings over cups of home-grown tea, the conversation that ranged from technical minutiae to theoretical extravaganzas to comments on the contemporary political scene, for the demonstration of the depths to which bad taste in clothing can sink, and for the truly hideous green, orange, and purple polyester dress that accompanied your last set of comments.

Great thanks go to all those who bravely served on my interdisciplinary Ph.D. program committee at a time when it was not at all clear where this research direction would go and whether it was capable of bearing fruit: Merrick Furst, Ronald Judy, Paul Hopper, Nancy Spivey, Kris Straub, and Dave Touretzky. I would particularly like to thank Ronald Judy for having done most of the legwork to get the program approved by the English department. His attention to the details of how the program should be structured and approved saved me much heartache and headache further down the road.

For help with this thesis, I would like to thank Paul Vanouse and Colin Piepgras, for advice on the design of the Industrial Graveyard; James Lester, for help with the design of the user study, which unfortunately never came into being; Bruce Blumberg, for general encouragement and especially for fed exing me a copy of the ALIVE video before my thesis proposal; Kerstin Dautenhahn, for special support and encouragement; and Bruce Blumberg, Rod Brooks, James Lester, and Luc Steels for permission to use their images in this thesis.

One of the greatest pleasures of being a graduate student in LCT is being immersed in an environment of smart, kooky, funky intellectuals whose greatest pleasure after a couple pints of Iron City is the thorough discussion of such issues as the structure of postmodern consciousness as reflected in contemporary advertising for feminine hygiene products. Thanks to my fellow grad students for the moral support, the stimulating and sometimes bizarre conversation, and the happening parties, some of which I will never forget and some of which I still can't remember. Special thanks are also due to honorary grad student Sharon Ghamari-Tabrizi, an inspiring teacher, ferocious intellect, and all-around target of my admiration.

In CS, I was fortunate to work in the context of the Oz Project, a tolerant and encouraging environment for work on the cutting edge of respectability. I am particularly grateful to Bryan Loyall, for many conversations about agent architectures and life; to Scott Neal Reilly, at once the most normal and the most twisted Oz-ite and therefore an infinite source of amusement; to Peter Weyhrauch, for an entire day in Toronto where he sang every sentence, opera-style; and to Michael Mateas, for letting me get to know him before he is really, really famous (I look forward to selling the tell-all memoir). I am also grateful to Catherine Copetas, for her competence in dealing with every problem, no matter how strange, I ever came to her with; and to Sharon Burks, for knowing all the rules and how to bend them in my favor.

Thanks to those who started me on the road to this thesis: to Hank Dardy, who trustingly gave me a research job in Computer Science when I barely knew how to program; to Simon Kasif, who herded me through a rigorous, theoretical undergraduate program, only to discover to his despair that I went off the deep end in graduate school; to Fritz Gutbrodt, who helped me put together the first inklings of what a synthesis of the humanities and technology could be like.

A graduate student's quality of life is extraordinarily affected by the officemates with whom s/he spends a large portion of his or her waking hours. I have been blessed to share my office with a group of inimitable characters and excellent friends: James Landay, the trash talker par excellence; Belinda Thom, the Queen B; and Dayne Freitag, international man of mystery. Y'all are the greatest!

I have been surrounded by many loving friends over my years at CMU. I am grateful to all of them for making these some of the best years of my life. I would especially like to thank the following people: Mary Tomayko, for many late-night phone calls and for being utterly unlike me and still my best friend; Kristin Nelson, for her pig-headedness, which

I adore, and for her sharp wit and willingness to tell it like it is; Jean Camp and Shaun McDermott, for being my second home and family; Nick Thompson, for being himself, an endless source of fascination and object of affection; Laura Ruetsche, for simultaneously being both the smartest and the least pretentious person I know, for her dry wit, and for her ability to blurt out the most unexpected comments at any tme; Stephanie Byram, for her unique and wise outlook on life and for glowing and exotic trip reports; Faye Miller, for her blistering forehand and for many companionable hours of thesis avoidance in her office in Wean Hall; Thorsten Joachims, for his steady nerves, for his unflagging enthusiasm for unexpected trips, unusual activities, and anything vaguely edible, and for his normally well-hidden bizarre side; Doug Davis, for a steady infusion of cheese and overstimulated intellect in my life; and Robbie Warner, for not being my kid and instead being my friend.

Contents

If you embrace the virtual life, don't do it mindlessly;
read what the best critics have to say.
— Howard Rheingold

Chapter 1

Introduction: Agents in Culture

Artificial Intelligence (AI) has come a long way. Particularly in the last ten years, the subfield known as 'agents' — artificial creatures that 'live' in physical or virtual environments, capable of engaging in complex action without human control — has exploded [Johnson, 1997] [Sycara and Wooldridge, 1998][1]. We can now build agents that can do a lot for us: they search for information on the Web [Shakes *et al.*, 1997], trade stocks [Analytix Inc., 1996], play grandmaster-level chess [Hsu *et al.*, 1990], patrol nuclear reactors [Baker and Matlack, 1998], remove asbestos [Schempf, 1995], and so on. Agents have come to be powerful tools.

But one of the oldest dreams of AI is the 'robot friend' [Bledsoe, 1986], an artificial being that is not just a tool but has its own life. Such a creature we want to talk to, not just to find out the latest stock quotes or the answer to our database queries, but because we are interested in its hopes and feelings. Yes, we can build smart, competent, useful creatures, but we have not built very many that seem complex, robust, and alive in the way that biological creatures do. Who wants to be buddies with a spreadsheet program, no matter how anthropomorphized? Somehow, in our drive for faster, smarter, more reliable, more useful, more profitable artificial agents, it seems like we may have lost something equally important: the dream of a creature which is, on its own terms, alive.

At the same time, as the notion of 'agent' has started to take on pop culture cachet, outside academics have begun to turn a not-always-welcome critical eye on the practices of AI. To humanists interested in how AI fits into broader culture, both the goals and the methodologies of AI seem suspect. With AI funding coming largely from the military and big business, critics may wonder if AI is just about building autonomous fighter pilots, more complex voicemail systems, and robots to replace human workers on assembly lines. The notion of the hyperrational, disembodied agent which still drives much AI research strikes many critics as hopelessly antiquated and even dangerous. AI research, these

[1] The format for citations in this thesis is as follows: [Smith, 1998] cites a particular work; ([Smith, 1998], 14) cites a particular page in a particular work; and (14) cites a particular page in the most recently mentioned work.

critics say, is about reproducing in silicon ideas of humanity that are hopelessly limited, leaving out much of what we value in ourselves. AI, in this view, is bad science and bad news.

These critiques, while not always equally easy for AI researchers to hear, could potentially help AI researchers develop better technical practices. They often focus on what has been left out of AI, helping us understand at a deep level why we have not yet achieved the AI dream of artificial creatures that are meaningfully alive, giving us a glimpse of the steps we could take towards fulfilling that dream, and advising us on integrating the practice of AI responsibly with the rest of life. Unfortunately, however, while being eloquent additions to such fields as anthropology, philosophy, or cultural studies, the critiques have often been unintelligible to AI researchers themselves. Lacking the context and background of humanist critics, researchers often see humanist concerns as silly or beside the point when compared to their own deep experiential knowledge of technology. Similarly, humanist critics have generally lacked the background (and, often, the motivation) to phrase their criticisms in ways that speak to the day-to-day technical practices of AI researchers. The result is the ghettoization of both AI and cultural critique: technical practices continue on their own course without the benefit of insight humanists could afford, and humanists' concerns about AI have little effect on how AI is actually done.

The premise of this thesis is that things can be different. Rather than being inherently antagonistic, AI and humanistic studies of AI in culture can benefit greatly from each other's strengths. Specifically, by studying AI not only as technology but also as a cultural phenomenon, we can find out how our notions of agents spring from and fit into a broader cultural context. Reciprocally, if the technology we are currently building is rooted in culturally-based ways of thinking, then by introducing new ways of thinking we can build new and possibly better kinds of technology.

This insight — that cultural studies of AI can uncover groundwork for new technology — forms the basis of this thesis. In particular, I look at methods for constructing artificial creatures that combine many forms of complex behavior. I analyze the technical state of the art with a cultural studies of science perspective to discover the limitations that AI has unknowingly placed upon itself through its current methodological presuppositions. I use this understanding to develop a new methodological foundation for an AI that can combine both humanistic and engineering perspectives. Finally, I leverage these insights in the development of agent technology, in order to generate agents that can integrate many behaviors while maintaining intentional coherence in their observable activity; or, colloquially speaking, appear more *alive*.[2]

But let's start at the beginning, with you and what you bring to this work. You may be an AI researcher, curious about the humanities or only interested in technology. You may be a cognitive scientist, a cultural critic, an anthropologist, a historian, an artist, all of these, some of these, none of these. You may be dying to know how to construct functional agents out of many behaviors; or you may be mildly curious about how AI has imported and modified methodologies from the industrial revolution. You may be a true believer in this interdisciplinary direction or you may

[2] What this means concretely will be made clear in Chapter 6.

be a die-hard skeptic.

In all these cases, this thesis has something to say to you, but in none of them can it do so without your help. This is a thesis which lives in the gap between two disciplines, AI and the cultural studies of science, which share almost nothing in their presuppositions, methodologies or values. As such, it is likely to please no one. If it is seen as a monolithic argument, to be accepted or rejected in its entirety, it will almost inevitably fail.

Instead, I would suggest that you try thinking of it as a toolbox of interconnected ways of thinking, each of which will be more or less useful to you depending on what you do now and what you want to use it for. If you can use the technology but find the philosophy on which it is based implausible, more power to you. If you appreciate the analysis of construction of knowledge about agents, but find the technical application deeply wrong-headed, that's OK too. But you will probably get the most out of this thesis if you find a way to make some sense of even the alien parts of this thesis.

In the rest of this introduction, I will try provide the background knowledge that you will likely need to feel at home in the rest of the thesis. I will introduce the fields of autonomous agents and cultural studies of science. I will give an overview of how agent research and broader culture are intimately intertwined. Then, I will explain how agent research and cultural studies have been profitably combined in the past, and how the approach for synthesizing them provided in this thesis grows out of these past traditions. This will set us up to delve into technical work in Chapter 2.

Introduction to Autonomous Agents

One of the dreams of AI is the construction of independent artificial beings. Rather than slavishly following our orders, or filling some tiny niche of activity that requires some aspect of intelligence (for example, playing chess), these artificial creatures would lead their own existences, have their own thoughts, hopes, and feelings, and generally be independent beings just as other people or animals are. In the 1950's and early 1960's, this dream for AI was embodied in cybernetics. For example, Walter built small robots with rudimentary "agenty" behaviors [Walter, 1963]. He called his robots 'turtles;' they would roam around their environment, seeking light, finding food, and avoiding running into things. Later models could do some rudimentary associative learning.

But as cybernetics fell out of fashion, AI research began to focus more on the cognitive abilities an artificial agent might need to have higher-level intelligence, and less on building small, complete (if not so smart) robots. At least partially because the task of reproducing a complete creature has been so daunting, AI spent quite a few years focused on building individual intelligent capabilities, such as machine learning, speech recognition, story generation, and computer vision. The hope was that, once these capabilities were generated, they could be combined into a complete agent; the actual construction of these agents was often indefinitely deferred.

More recently, however, the field of autonomous agents has been enjoying a renaissance. The area of autonomous agents focuses on the development of programs that more closely approach representations of a complete person or creature. These agents are programs which engage in complex activity without the intervention of another program or person. Agents may be, for example, scientific simulations of living creatures [Blumberg, 1994], characters in an interactive story [Bates, 1994], robots who can independently explore their environment [Simmons *et al.*, 1997], or virtual 'tour guides' that accompany users on their travels on the World Wide Web [Joachims *et al.*, 1997].

While these applications vary wildly, they share the idea that the program that underlies them is like a living creature in some important ways. Often these ways include being able to perceive and act on their (perhaps virtual) environment; being autonomous means they can make decisions about what to do based on what is happening around them and without necessarily consulting a human for help. Agents are also often imputed with rationality, which is defined as setting goals for themselves and achieving them reasonably consistently in a complex and perhaps hostile environment.

Agent as Metaphor

The definition of what exactly is and is not an agent has at times been the source of hefty controversy in the field. Mostly these controversies revolve around the fact that any strictly formal definition of agenthood tends to leave out such well-beloved agents as cats or insects, or include such items as toasters or thermometers that a lay person would be hard-pressed to call an agent. With some of the looser definitions of agents, for which the word 'agent' just seems to be a trendy word for 'program,' skeptics can be forgiven for wondering why we are using this term at all.

In this thesis, I will take agenthood broadly to be a sometimes-useful way to frame inquiry into the technology we create. Specifically, agenthood is a metaphor we apply to computational entities we build when we wish to think of them in ways similar to the ways we understand living creatures. Calling a program an agent means the program's designer or the people who use it find it helpful or important (or, for that matter, attractive to funders to think of the program as an independent and semi-intelligent coherent being. For example, when we think of our programs as agents we focus our design attention on 'agenty' attributes we would like the program to have: the program may be self-contained; it may be situated in a specific, local environment; it may engage in 'social' interactions with other programs or people.[3] When a program is presented to its user as an agent, we are encouraging the user to think of it not as a complex human-created mechanism but as a user-friendly, intelligent creature. If 'actually' some kind of tool, the creature is portrayed as fulfilling its tool-y functions by being willing to do the user's bidding [Lanier, 1996] [Wise, 1996]. Using the metaphor 'agent' for these applications lets us apply ideas about what living agents such as dogs, beetles, or bus drivers are like to the design and use of artificially-created programs.

[3] I am indebted to Filippo Menczer for this observation.

Agenthood in Classical and Alternative AI

But not all AI researchers agree on which conceptions of living agents are appropriate or useful for artificial agents. The past 10 years in particular have seen an at times spectacular debate between different strains of thought about the proper model of agent to use for AI research (see e.g. [Cognitive Science, 1993]). Rodney Brooks [Brooks, 1990], for instance, divides the field into symbolically-grounded vs. physically-grounded agents. Agents based on symbols work by manipulating representations of their environment; physically-based agents work by manipulating and reacting to the environment itself. Philip Agre and David Chapman [Agre and Chapman, 1990] distinguish agents using 'plans-as-programs' from agents using 'plans-as-communication;' they divide programs into ones that engage in abstract, hierarchical planning of activity before engaging in it (often including formal proofs that the plan will fulfill the goal the agent is given) versus ones that are designed to take advantage of an action loop with respect to their environment and may only refer to plans as ways to structure common activities. Another common distinction is between situated and cognitive agents; situated agents are thought of as embedded within an environment, and hence highly influenced by their situation and physical make-up, whereas cognitive agents engage in most of their activity at an abstract level and without reference to their concrete situation.

These divisions are not independent; rather, they tend to repeat similar categories with different names. Specifically, these rubrics tend to organize themselves into two conceptual clusters: a main stream often termed *classical* AI (also known as Good Old-Fashioned AI, cognitivistic AI, symbolic cognition, top-down AI, knowledge-based AI, etc.) and an oppositional stream we can term *alternative* AI (also known as new AI, nouvelle AI, ALife, behavior-based AI, reactive planning, situated action, bottom-up AI, etc.).[4] Not every AI system neatly falls into one or the other category — in fact, few can be said to be pure, unadulterated representatives of one or other. But each stream represents a general trend of thinking about agents that a significant number of systems share.

For AI researchers, the term classical AI refers to a class of representational, disembodied, cognitive agents, based on a model that proposes, for example, that agents are or should be fully rational and that physical bodies are not fundamentally pertinent to intelligence. The more extreme instances of this type of agent had their heyday in the 60's and 70's, under a heady aura of enthusiasm that the paradigms of logic and problem-solving might quickly lead to true AI. One of the earliest examples of this branch of AI is Allen Newell and Herbert Simon's GPS, the somewhat optimistically titled "general problem solver." This program proceeds logically and systematically from the statement of a mathematical-style puzzle to its solution [Newell and Simon, 1972]. Arthur Samuel's checker player, one of the first programs that learns, attempts to imitate intelligent game-playing by learning a polynomial function to map aspects of the current board state to the best possible next move [Samuel, 1995]. Terry Winograd's SHRDLU maintains a simple representation of blocks lying on a table, and uses a relatively

[4] For similar analyses, see e.g. [Steels, 1994] [Varela *et al.*, 1991] [Brooks, 1990] [Norman, 1993].

straightforward algorithm to accept simple natural language commands to move the virtual blocks [Winograd, 1972]. While the creators of these programs often had more subtle understandings of the nature of intelligence, the programs themselves reflect a hope that simple, logical rules might underlie all intelligent behavior, and that if we could discover those rules we might soon achieve the goal of having intelligent machinery.

But the classical model, while allowing programs to succeed in many artificial domains which humans find difficult, such as chess, unexpectedly failed to produce many behaviors humans find easy, such as vision, navigation, and routine behavior. The recognition of these failures has led to a number of responses in the 80's and 90's. Some researchers — most notably Winograd, who wrote an influential book with Fernando Flores on the subject [Winograd and Flores, 1986] — have decided that the intellectual heritage of AI is so bankrupt they have no choice but to leave the field. By far the majority of AI researchers have remained in a tradition that continues to inherit its major research framework from classical AI, while expanding its focus to try to incorporate traditionally neglected problems (we might call this 'neo-classical AI'). A smaller but noisy group has split from classical AI, claiming that the idea of agents that classical AI tries to promote is fundamentally wrong-headed.

These researchers, who we will here call alternative AI, generally believe that the vision of disembodied, problem-solving minds that explicitly or implicitly underlies classical AI research is misguided. Alternative AI focuses instead on a vision of agents as most fundamentally nonrepresentational, reactive, and situated. Alternative AI, as a rubric, states that agents are situated within an environment, that their self-knowledge is severely limited, and that their bodies are an important part of their cognition.

Technology as Theory of Subjectivity

The dialogue and debate between these two types of agents is not only about a methodology of agent-building. An underlying source of conflict is about which aspects of being human are most essential to reproduce. Classicists do not deny that humans are embodied, but the classical technological tradition tends to work on the presupposition that problem-solving rationality is one of the most fundamental defining characteristic of intelligence, and that other aspects of intelligence are subsidiary to this one. Likewise, alternativists do not deny that humans can solve problems and think logically, but the technology they build is based on the assumption that intelligence is inherent in the body of an agent and its interactions with the world; in this view, human life includes problem-solving, but is not a problem to be solved.

It is in these aspects of AI technology — ones that are influenced by and in turn influence the more philosophical perspectives of AI researchers — that we can uncover, not just the technology of agents, but also theories of agenthood. Two levels of thought are intertwined in both these approaches to AI: (1) the level of day-to-day technical experience, what works and what doesn't work, which architectures can be built and which can't; and (2) the level of background philosophy — both held from the start and slowly and mostly unconsciously imbibed within the developing technical traditions — which underlies the way in which the whole complex and undefined conundrum of recreating life in the computer is understood. Running through and along with the technical arguments are more philosophical arguments about what human life is or should be like, how we can come to understand it, what it means to be meaningfully alive.

Technical researchers may feel uncomfortable making this connection between technology and fuzzier ideas about what it is to be human. But this is not revolutionary; the connection between living and artificial agent is ingrained in AI, allowing the connection between AI and psychology that forms cognitive science. For example, when Newell describes his concept of the 'knowledge level' — a way of comprehending beings as agents, rather than as physical organisms or computers — he means for this way of thinking to describe *both* artificial *and* living agents [Newell, 1981]. Both these kinds of agents are described using the same kind of structure: "an agent is composed of a set of actions, a set of goals, and a body... [T]he agent processes its knowledge to determine the actions to take. Finally, the behavior law is the *principle of rationality*. Actions are selected to attain the agent's goals" (13). For Newell, at the knowledge level, an agent is defined to consist of actions, goals, and body; for an entity to be considered an agent, its actions must be oriented to achieving goals. These attributes of agents are considered to hold whether we are talking about computers or people. The knowledge-level theory implies that both kinds of agents are fundamentally structured so that their behavior consists of rational attempts to achieve plausible goals.[5]

But even researchers who do not claim to be doing cognitively plausible work draw their inspiration in part from theories of living agents. This is demonstrated, for example, by the very title of Brooks' position paper opposing classical AI, *Elephants Don't Play Chess*. While Brooks does not claim to be building structures isomorphic to ones inside the mind, he does think that considerations of what 'real' agents do in the world are part of the consideration that should go into the design of an alternative agent. Here, he claims that rational, symbolic, problem-solving behavior is inessential to an agent's existence in the world, which is rather dominated by the need for perception and reactivity.

Cultural theorists use the term 'subjectivity' to refer to theories or models of consciousness. A theory of subjectivity suggests what existence is like, how we come to experience ourselves and the world around us, what it feels like or means to be a person. From the previous discussion, it seems clear that AI includes not only conflicting theories of technology but also, implicitly, conflicting theories of subjectivity. Specifically, classical AI technology is based on a model of

The term 'subjectivity' is related to the perhaps more familiar term 'subjective' in that they both refer to personal experience. 'Subjective' knowledge is something that is known to you as an individual, whereas 'objective' knowledge can be thought of as something that would hold true for anyone, and is therefore not related to or dependent on your life experience.

[5] By 'rationality,' AI researchers often mean 'bounded rationality,' i.e. that the rationality of an agent's behavior is limited to its (presumably limited) knowledge. What I mean to get at here is *not* that the knowledge-level theory implies that computers and people are hyperrational (and perhaps by extension hyperintelligent). Rather, I argue that setting up rationality as one of the *fundamental* characteristics by which agentiness can be defined means that agents which are behaving *irrationally* (as humans often do) are *flawed* in their agenthood. Of course, this irrationality can be, and in AI often is, redefined as rationality with flawed knowledge or in the pursuit of perverted goals, such as that a person is, for example, rationally trying to harm him- or herself — a redefinition that handily circumvents having to deal with the still unanswered (and perhaps unanswerable) question of whether rationality should be considered a fundamental, defining property of the experience of being in the world.

consciousness as essentially representational, rational, and disembodied. Alternative AI technology presupposes that it is essentially reactive, situated, and embodied.

These two categories can be clearly seen within AI research. Within that research community, they are generally seen as coming about from certain tensions in technical practice itself. Interestingly, they correspond closely to two categories cultural theorists regularly employ to talk about historical notions of what it means to be a person: *rational* and *schizophrenic* subjectivity.[6]

Rational subjectivity refers to a common way of conceiving humanity in the West since the Enlightenment. It is historically anchored in the work of René Descartes, the Enlightenment philosopher who derives proof of his existence from the fact that he thinks. Rational subjectivity is based on this Cartesian focus on logical thought: the mind is seen as separated from the body, it is or should be fundamentally rational, and cognition divorced from emotion is the important part of experience. This model has overarching similarities with, for instance, Allen Newell's theory of Soar, which describes an architecture for agents that grow in knowledge through inner rational argumentation [Newell, 1990]. Most models built under Soar are focused on how this argumentation should take place, leaving out issues of perception and emotion (though there are certainly exceptions; see e.g. [Pearson *et al.*, 1993]).

The development of the notion of schizophrenic subjectivity is based on perceived inadequacies in the rational model, and is influenced by but by no means identical to the psychiatric notion of schizophrenia (we will discuss this relationship in more detail in Chapter 2). While rational subjectivity presupposes that people are fundamentally or optimally independent rational agents with only tenuous links to their physicality, schizophrenic subjectivity sees people as fundamentally social, emotional, and bodily. It considers people to be immersed in and to some extent defined by their situation, the mind and the body to be inescapably interlinked, and the experience of being a person to consist of a number of conflicting drives that work with and against each other to generate behavior. In AI, this form of subjectivity is reflected in Brooks's subsumption architecture, in which an agent's behavior emerges from the conflicting demands of a number of loosely coupled internal systems, each of which attempts to control certain aspects of the agent's body based almost entirely on external perception rather than on internal cogitation [Brooks, 1986a].

Each class of agent architectures closely parallels a model of subjectivity. Just as alternative AI has arisen in an attempt to address flaws in classical AI, the schizophrenic model of subjectivity has arisen in response to perceived flaws in the rational model's ability to address the structure of contemporary experience. Each style of agent architecture shows a striking similarity to a historical model of subjectivity that cultural theorists have identified.

This close relationship between a technical debate in a subfield of computer science and philosophical trends in Western culture as a whole may come as a surprise. But a moment of reflection reveals where the

[6]This idea is a more common observation among cultural theorists who study AI. See, for example, [Barton, 1995] and [de Mul, 1997].

connection lies. AI researchers are also human beings, and as such inhabit and are informed by the broader society that cultural theorists study. From this point of view, AI is simply one manifestation of culture as a whole. Its technical problems are one specific arena where the implications of ideas that are rooted in background culture are worked out.

But if AI is fundamentally embedded in and working through culture, then cultural studies and AI may have a lot to say to each other. Specifically, the practitioners of cultural studies — who we will here refer to as cultural critics or cultural theorists — have spent a lot of time thinking about and debating subjectivity. AI researchers have spent a lot of time thinking about and debating architectures for autonomous agents. Once these two are linked, each body of work can be used to inform the other. If agents use a particular theory of subjectivity, then we can use ideas about this theory to inform our work on agents. And if agents are a manifestation of a theory of subjectivity, then studying these agents can give us a better idea of what that theory means. In order to make this idea concrete, we will now look at cultural studies and its relationships to science in more detail.

Cultural Studies Meets Science

Cultural studies — and its related philosophy, cultural theory — is a hybrid collection of literary scholars, anthropologists, philosophers, sociologists, historians, and other sympathetic humanists. While cultural theorists are heterogeneous in both method and philosophy, they generally aim to understand human experience as it is formed and expressed through a variety of cultural forms. A common interest of cultural theorists is understanding how the structure of society both constrains and enables human understanding of ourselves and each other.

One way of understanding the mindset of cultural studies is by looking at how it has grown out of literary studies. Literary studies originally were confined to high literature, i.e. stories by writers such as Shakespeare who are acknowledged as great. Over time, literary scholars began to apply the methods of literary studies to 'low' literature as well, for example dime store novels, as well as works by authors outside the main traditions of Western culture. Soon, these scholars noticed that the same techniques also worked for film, leading to film studies. Gradually, the field expanded to cover all forms of cultural production, including television, advertising, law, politics, religion, and science.

The cultural studies of science — also termed cultural critique of science or science studies — aim to understand science as it relates to the culture of which it is a part. It broadly functions as a kind of 'science criticism' analogous to literary criticism [Harding, 1994]: one of its major goals is to understand and improve the quality and relevance of scientific work by thinking about how it stems from and affects the rest of culture. Science, too, has an ideal of improving itself by continuously subjecting itself to rigorous self-criticism [Rouse, 1993]. But like philosophy of science, science studies aims to understand and improve not only particular technical work, but also the very mechanisms by which science works and through which it produces knowledge. Science studies goes beyond both science and philosophy of science by relating scientific methodology and assumptions to other cultural practices which many scientists and philosophers see as external or irrelevant to the production of science.

Science Wars

Science studies examines culturally-based metaphors that inform scientific work, and thereby often uncovers deeply-held but unstated assumptions that underly it. Scientists are also generally interested in understanding the forces, both conscious and unconscious, that can shape their results. If there are ways in which they can better understand the phenomena they study or build the technology they want to create, they are all ears. In this respect, the insights of science studies can contribute great value to science's self-understanding [Keller, 1985].

At the same time, many practitioners of science studies are deeply interested in science as it is actually practiced on a day-to-day level. This means scientists, with their in-depth personal experience of what it means to do scientific work, are privy to perspectives that can enrich the work of their science studies counterparts. Science studies simply is not possible without science, and an important component of it is an accurate reflection of the experiences of scientists themselves.

With all the advantages that cooperation could bring, you might think that science and science studies would be enthusiastic partners on the road to a shared intellectual enterprise. Alas, that is far from the case! Unfortunately, productive exchanges between cultural critics and scientists interested in the roots of their work are hampered by the disciplinary divide between them [Snow, 1969]. This divide blocks cultural critics from access to a complete understanding of the process and experience of doing science, which can degrade the quality of their analyses and may lead them to misinterpret scientific practices. At the same time, scientists have difficulty understanding the context and mindset of critiques of their work, making them unlikely to consider such critiques seriously or realize their value for their work, potentially even leading them to dismiss all humanistic critiques of science as fundamentally misguided [Gross and Levitt, 1994].

This feedback loop of mutual misunderstanding has grown into a new tradition of mutual kvetching. Cultural critics may complain that scientists unconsciously reproduce their own values in their work and then proclaim them as eternal truth. They may feel that scientists are not open to criticism because they want to protect their high (relative to the humanities') status in society. Simultaneously, scientists sometimes complain that cultural critics are absolute nihilists who do not believe in reality and equate science with superstition.[7] They fear that cultural critics undermine any right that science has as a source of knowledge production to higher status than, say, advertising. Finally, both sides complain incessantly — and correctly —- of being cited, and then judged, out of context.

The unfortunate result of this situation is a growing polarization of the two sides. In the so-called "Science Wars" [Social Text, 1996], pockets

[7] This is exacerbated by the fact that the notion of 'reality' used by many scientists in their criticism of science studies does not bear much relation to the long and deep tradition of the usage of that term in cultural studies of science.

of fascinating interdisciplinary exchanges and intellectually illuminating debate are sadly overwhelmed by an overall lack of mutual understanding and accompanying decline of goodwill. While most participants on both sides of the divide are fundamentally reasonable, communication between them is impaired when both sides feel misunderstood and under attack. This siege mentality not only undermines the possibility for productive cooperation; with unfortunate frequency, it goes as far as cross-fired accusations of intellectual bankruptcy in academic and popular press and nasty political battles over tenure. These unpleasant incidents not only help no one but also obscure the fact that *both* the academic sciences *and* the humanities are facing crises of funding in an economy that values quick profit and immediate reward over a long-term investment in knowledge. In the end, neither science nor science studies benefits from a situation best summed up from *both* sides by Alan Sokal's complaint: "The targets of my critique have by now become a self-perpetuating academic subculture that typically ignores (or disdains) reasoned criticism from the outside" [Sokal, 1996].[8]

AI Skirmishes

While most scientists remain blissfully unaware of the Science Wars, they are not unaffected by them. Within AI, the tension between the self-proclaimed defenders of scientific greatness and the self-identified opponents of scientific chauvinism is worked out under the table. In particular, the sometimes tendentious clashes between classical and alternative AI often reflect arguments about science and the role of culture in it.

This can be seen most clearly in a rather unusual opinion piece that appeared several years ago in the *AI Magazine* [Hayes *et al.*, 1994] . The remarkable rhetoric of this essay in a journal more often devoted to the intricacies of extracting commercially relevant information from databases may be appreciated in this excerpt:

> Once upon a time there were two happy and healthy babies. We will call them Representation Baby (closely related to Mind Baby and Person Baby) and Science Baby (closely related to Reality Baby).
>
> These babies were so charming and inspirational that for a long time their nannies cared for them very well indeed. During this period it was generally the case that ignorance was pushed back and human dignity increased. Nannies used honest, traditional methods of baby care which had evolved during the years. Like many wise old folk, they were not always able to articulate good justifications for their methods, but they worked, and the healthy, happy babies were well growing and having lots of fun.
>
> Unfortunately, some newer nannies haven't been so careful, and the babies are in danger from their zealous ways. We will focus on two nannies who seem to be close friends and

[8] Alan Sokal happens to be a physicist complaining about science studies, but this quote works just as aptly to summarize the complaints made the other way around.

> often can be seen together - Situated Nanny (called SitNanny for short) and Radical Social Constructivist Nanny (known to her friends as RadNanny) (15).[9]

A little decoding is in order for those not intimately aware of both the AI debates and the Science Wars. “SitNanny” represents situated action, a brand of alternative AI that focuses its attention on the way in which agents are intimately related to, and cannot be understood without, their environment. “RadNanny” is the embodiment of the cultural studies of science, social constructivism being the belief that science, like every other human endeavor, is at least partially a product of sociocultural forces (the ‘radical’ here functions as little more than an insult, but implies that science is *purely* social, i.e. has absolutely no relationship to any outside reality).

Having broken the code, the implication of this excerpt is clear: everything in AI was going fine as long as we thought about things in terms of science and knowledge representation. Of course, this science was not always well-thought-out, but it was fundamentally good. That is, until that dastardly alternative AI came along with cultural studies in its tow and threatened nothing less than to *kill the babies*.

> Note for the non-AI readers: knowledge representation can be thought of as the belief that AI agents have explicit representations of the outside world in their head, which they manipulate in order to forecast what affect their actions will have on the world.

Now any cultural critic worth his or her salt will have some choice commentary on a story in which the positive figures are all male babies living the life of leisure, and the negative figures all lower-class working women.[10] But the really interesting rhetorical move in this essay is in the alignment of the classical-alternative AI debate with the Science Wars. Classical AI, we learn, is good science. Alternative AI, while having some good ideas, is dangerous, among other reasons because it is watering down science with other ideas: “concepts from fringe neurology, sociology, ethnomethodology, and political theory; precomputational psychological theory; and God knows what else” (19). Alternative AI is particularly dangerous because it believes that agents cannot be understood without reference to their environment. Hence, it is allied with the “cult” (20) of science studies, which believes that scientists cannot be understood without reference to their sociocultural environment.

Since the majority of their audience presumably has little awareness of science studies, the authors are happy to do their part for interdisciplinary awareness by explaining what it is. They state, in a particularly nice allusion to 1950’s anti-Communist hysteria, that science studies aims at nothing less than to “reject the entire fabric of Western science” (15). Science studies, we are informed, believes “that all science is arbitrary and that reality is merely a construction of a social game” (23). In the delightful tradition of the Science Wars, several quotations are taken out of context to prove that cultural critics of science believe that science is merely an expendable myth.

The statements Hayes et. al. make are simply inaccurate descriptions of science studies. In reality, science studies tends to be agnostic on such questions as the arbitrariness of science and on the nature of reality, to

[9] This excerpt cannot, however, carry the full force of the original, which contains several full-page 19th-century woodcuts displaying suffering babies and incompetent or evil nannies (labeled, for example, “The Notorious RadNanny Looking For Babies”).

[10] One must presume that the authors were aware of this and did their best to raise cultural critics’ hackles.

which science studies generally does not claim to have any more access than science does. When science studies *does* look into these issues it does so in a much more subtle and complex way than simply rejecting or accepting them.

But what is more important than these factual inaccuracies is that the article promotes the worst aspects of the Science Wars, since the very *tone* of the article is chosen to preclude the possibility of productive discussion. Science studies is simply dismissed as ludicrous. If uninformed scientists reading the article have not by the end concluded that science studies is an evil force allied against them, with alternative AI its unfortunate dupe, it is certainly not for lack of trying.

AI in Culture, AI as Culture

But is it really true that science studies is an enemy of AI? After all, no one disputes that AI is, among other things, a social endeavor. Its researchers are undeniably human beings who are deeply embedded in and influenced by the social traditions in which they consciously or unconsciously take part, including but by no means limited to the social traditions of AI itself. It seems that taking these facts seriously might not necessarily damage AI, but could even help AI researchers do their work better.

In this section, we will buck the trend of mutual disciplinary antagonism by exploring the potential of what Agre calls *critical technical practices* [Agre, 1997]. A critical technical practice is a way of actually doing AI which incorporates a level of reflexive awareness of the kind espoused by science studies. This may include awareness of the technical work's sociocultural context, its unconscious philosophies, or the metaphors it uses. We will look at various AI researchers who have found ideas from science studies helpful in their technical work. With this previous work as its basis, in the rest of this chapter I will explain the approach to synthesizing AI and cultural studies I am taking in this thesis.

A Short History of Critical Technical Practices

From the rather heated rhetoric of the Science Wars, you might be tempted to think that science and science studies have nothing of value to share with each other. Often, voices on the 'pro-science' side of the debate say that the cultural studies of science has no right to speak about science because only scientists have the background and ability to understand what science is about and judge it appropriately. At the same time, the 'pro-culture' side of the debate may feel that scientists neither know about nor care to ameliorate the social effects of their work.[11]

[11] The way in which these attitudes cut off communication is not infrequently illustrated to me in the flesh. For example, a cultural theorist who was once introduced to me immediately said, "So, you work in AI. How does it feel to be the instrument of global capital in the replacement of workers by machinery?" I immediately responded, "I don't know. How does it feel to be the instrument of the university in the training of the next generation of happy materialist consumers?" — not because the question was unreasonable, but because its very phrasing demonstrated that the possibility for meaningful communication had been deliberately closed off from the start. Lest cultural theorists be singled out for judgment, in my experience scientists are quite capable of similar 'conversations.'

These attitudes can only be maintained by studiously avoiding noticing the people who are *both* scientists *and* cultural critics. Gross and Levitt's influential onslaught against science studies, for example, argues that cultural critics are irresponsible and dangerous because they are ignorant of the science they criticize. This argument is made easier by counting interdisciplinarians who do *both* science *and* cultural studies as (good, responsible) scientists and not as (bad, irresponsible) cultural critics (the question of why those scientists would find it interesting or even fruitful to keep such unseemly company is left unanswered) [Gross and Levitt, 1994]. And in an exhaustive survey of every important figure in cultural studies, some of the most influential 'culturalist scientists' are left out together. A glaring omission is Richard Lewontin, whose influential books on the cultural aspects of biology are the sidelight to an illustrious career as a geneticist [Levins and Lewontin, 1985] [Lewontin *et al.*, 1984].[12]

Similarly, the hypothesis that scientists do not know or care about the effects of their work is contradicted by the work of Martha Crouch [Crouch, 1990]. Crouch is a botanist who, after many years of research, noticed that the funding of botany combined in practice with the naive faith of scientists in their own field to completely undermine the idealistic goals of plant scientists themselves. Crouch determined to help scientists such as herself achieve their own stated goals of, for example, feeding the hungry, by adding to their self-understanding through the integration of cultural studies with botany.

But, to be fair, much of the work integrating science with science studies may be invisible to both cultural critics themselves and the scientists whose form of intellectual output seems to largely be attacks on those on the other side of the great intellectual divide. This is because scientists who are actually using culturalist perspectives in their work generally address that work to their scientific subcommunity, rather than to all of science and science studies as a whole. And in work that is addressed to a technical subfield, it is usually not particularly advantageous to mention that one's ideas stem from the humanities, particularly if they come from such unseemly company as hermeneutics, feminism or Marxism.

Here, we will uncover the history of the use of culturalist perspectives within AI as a part of technical work. It turns out that within AI, the use of the humanities is not just a couple of freak accidents traceable to a few lone geniuses and / or lunatics. Rather, there is a healthy if somewhat hidden tradition of a number of generations of AI researchers who have drawn inspiration from the humanities in ways that have had substantial impact on the field as a whole. We will be interested both in finding out how cultural studies was found to be useful, and in the concrete methods various researchers have used to combine the fields.

Winograd and Flores

Terry Winograd is one of the first and certainly one of the most notorious in his usage of critical theory to analyze AI from the AI researcher's point of view. As mentioned in the review of classical AI, Winograd was a well-known researcher into the machine generation of human language.

[12] For Lewontin's roasting response to Gross and Levitt, see [Lewontin, 1995].

Upon collaboration with Fernando Flores, an economist, Winograd became interested in Heidegger. In an unexpected move, after trying to understand AI in a Heideggerian sense Winograd chose to jettison AI altogether as impossible.

In [Winograd and Flores, 1986], Winograd and Flores describe AI as fundamentally too invested in the analytic tradition to ever be able to address fundamental attributes of intelligence. In particular, they focus on the Heideggerian idea that a person in the world is always operating from a set of background prejudices that can not be finitely, hence mechanically, articulated. AI *can* solve problems that are formally specified and circumscribed, but will always fail to attain true intelligence because "[t]he essence of intelligence is to act appropriately when there is no simple pre-definition of the problem or the space of states in which to search for a solution" (98).

Prejudice here refers to things that you subconsciously believe without having justified them, as opposed to negative stereotypes of people different from yourself.

While Winograd and Flores's arguments certainly made a splash in the field, it must be honestly stated that they probably did not cause too many scientists to leave AI (and they were not intended to). The basic flaw from this perspective in the argument is that it forces AI researchers to choose between believing in Heidegger and believing in AI. One can hardly blame them if they stay with the known evil.

What is interesting to those who remain in AI, however, is Winograd and Flores's methodology for combining a philosophical perspective with AI. Winograd and Flores analyze the limitations of AI that stem from its day-to-day methodologies. When they find those constraints to exclude the possibility of truly intelligent behavior, they decide instead to start building systems in which those constraints become strengths. In other words, they decide that artificial systems necessarily have certain characteristics of rigidity and literalness, then ask themselves what sorts of social situations could be aided by a rigid, literal system. They then build a system that is an enforcer of social contracts in certain, limited situations where they feel it is important that social agreements be clearly delineated and agreed upon. Specifically, the system articulates social agreements within work settings, so that workers are aware of who has agreed to do what. This new system is designed to be useful precisely because of the things that were previously limitations! Winograd and Flores, then, use cultural studies to inform technical development by finding constraints in its methodologies, and then using those constraints so that they become strengths.

Suchman

Lucy Suchman is an anthropologist who, for a time, studied AI researchers and, in particular, the ideas of 'planning' [Suchman, 1987]. Planning is an area of AI that is, at its most broad, devoted to deciding what to do. Since this broad conception does not really help you sink your teeth into the problem, a more limited notion has been generally used in AI. This concept of planning is a type of problem-solving where an agent is given a goal to achieve in the world, and tries to imagine a set of actions that can achieve that goal, generally by using formal logic.

Suchman noticed that the ideas of planning were heavily based on largely Western notions of, among other things, route planning. She then asked herself what kind of 'planning' you would have if you used the notions of a different society. By incorporating perspectives from Micronesian society, she came up with the concept of 'situated action,' which you may remember as the butt of ridicule in Hayes et. al.'s "On Babies and Bathwater" (page 11).

Situated action's basic premise is to generate behavior on the fly according to the local situation, instead of planning far ahead of time. Although Suchman herself made no claims to technical fame, her ideas became influential among AI researchers who were working on similarly-motivated technology (see below), becoming an important component in an entire subfield AI researchers now either love or hate, but generally cannot ignore. Her methodology, in sum, is to notice the culture-boundedness of a particular metaphor ("planning") that informs technical research, then ask what perspectives a very different metaphor might bring to the field instead. The point in her work is not that Western metaphors are 'wrong' and non-Western ones are 'right,' but that new metaphors can spawn new machinery that might be interesting in different ways from the old machinery.

Chapman

David Chapman was a graduate student at MIT when together with Agre, whose work is described separately below, he developed an agent architecture that was heavily influenced by Suchman's ideas, as well as by ethnomethodology [Chapman, 1990]. This architecture is described in more detail in Chapter 2. Chapman's contribution in this history of interdisciplinary methodologies in AI is his articulation of the value of 'ideas' — as opposed to proofs or technical implementation — in technical practice.

Chapman argues that some of the most interesting papers in AI do not make technical contributions in any strict sense of the term — i.e., that the best methodology for AI is not necessarily that of empirical natural science. "[Some of the best] papers prove no theorems, report no experiments, offer no testable scientific theories, propose technologies only in the most abstract terms, and make no arguments that would satisfy a serious philosopher.... [Instead, t]hese papers have been influential because they show us powerful ways of thinking about the central issues in AI" (214). Suchman's anthropological work in AI is a living example in Chapman's work of such an influential idea.

Agre

Of all AI researchers, Agre has probably done the most extensive and explicit integration of critical viewpoints with AI technology. In his thesis, for example, Agre integrates ethnomethodology with more straightforward AI techniques [Agre, 1988]. He uses ideas from ethnomethodology both to suggest what problems are interesting to work on (routine behavior, instead of expert problem-solving) and to suggest technical solutions (deictic, or subjective representation instead of objective representation).

Together with Chapman, Agre uses a philosophical approach influenced by Winograd's Heideggerian analysis of AI, but based more primarily on the work of such ethnomethodologists as Suchman and Garfinkel, to develop not only a new methodology for building agents, but also a new understanding of what it means to be an agent in the world that goes beyond views of life as consisting of rational problem-solving.

> The world of everyday life... is not a problem or a series of problems. Acting in the world is an ongoing process conducted in an evolving web of opportunities to engage in various activities and contingencies that arise in the course of doing so.... The futility of trying to control the world is, we think, reflected in the growing complexity of plan executives. Perhaps it is better to view an agent as *participating* in the flow of events. An embodied agent must *lead a life*, not *solve problems* ([Agre and Chapman, 1990], 20).

This re-understanding of the notion of agent has been an important intellectual strand in alternative AI's reconceptualization of agent subjectivity.

In recent work, Agre has distilled his approach to combining philosophy, critical perspectives, and concrete technical work into an articulated methodology for critical technical practices per se. Agre sees critical reflection as an indispensable tool in technical work itself, because it helps technical researchers to understand in a deep sense what technical impasses are trying to tell them. He sums up his humanistic approach to AI with these postulates:

> 1. AI ideas have their genealogical roots in philosophical ideas. 2. AI research programs attempt to work out and develop the philosophical systems they inherit. 3. AI research regularly encounters difficulties and impasses that derive from internal tensions in the underlying philosophical systems. 4. These difficulties and impasses should be embraced as particularly informative clues about the nature and consequences of the philosophical tensions that generate them. 5. Analysis of these clues must proceed outside the bounds of strictly technical research, but they can result in both new technical agendas and in revised understandings of technical research itself. [Agre, 1995]

Humanists will recognize Agre's methodology as a kind of hermeneutics, i.e. a process of interpretation that goes beyond surface appearances to discover deeper meanings. For Agre, purely technical research is the surface manifestation of deeper philosophical systems. While it is certainly possible for technical traditions to proceed without being aware of their philosophical bases, technical impasses provide clues that, when properly interpreted, can reveal the philosophical tensions that lead to them. If these philosophical difficulties are ignored, chances are that technical impasses will proliferate and remain unresolved. If, however, they are acknowledged, they can become the basis for a new and richer technical understanding.

In [Agre, 1997], Agre develops a methodology for integrating AI and the critical tradition through the use of deconstruction. Deconstruction is a technique developed by philosopher Jacques Derrida for analyzing texts in order to bring out inherent contradictions hidden in them [Derrida, 1976] [Culler, 1982]. Agre's methodology involves the following steps:

Dear humanists, forgive me for this reductive explanation, but *you* try explaining deconstruction to engineers in one sentence or less.

1. Find a metaphor that underlies a particular technical subfield. An example of such a metaphor is the notion of disembodiment that underlies classical AI.

2. Think of a metaphor that is the *opposite* of this metaphor. The opposite of disembodied agents would be agents that are fundamentally embodied.

3. Build technology that is based on this opposite metaphor. Embodied agents are an essential component of Rod Brooks's groundbreaking work, which is described in more detail in Chapter 2. This technology will inevitably have both new constraints and new possibilities when compared to the old technology.

In Agre's work, metaphorical analysis can become the basis for widening our perspective on the space of possible technologies.

Varela, Thompson, and Rosch

Francisco Varela, Evan Thompson, and Eleanor Rosch do not combine AI with cultural studies. Varela is a well-known cognitive scientist (a sister discipline of AI); Thompson and Rosch are philosophers. Nevertheless, their work is closely related to syntheses of AI and the humanities and deserves to be addressed along with them.

In [Varela *et al.*, 1991], Varela, Thompson and Rosch integrate cognitive science with Buddhism, particularly in the Madhyamika tradition. They do this by connecting cognitive science as the science of cognition with Buddhist meditation as a discipline of experience. Current trends in cognitive science tend to make a split between cognition and consciousness, to the point that some cognitive scientists call consciousness a mere illusion. Instead, Varela et. al. connect cognition and experience so cognitive scientists might have some idea of what their work has to do with what it means to be an actual, living, breathing human being.

Varela, Thompson, and Rosch stress that cognitive science — being the study of the mind — should be connected to our actual day-to-day experience of what it means to have a mind. What they mean here by experience is not simple existence per se but a deep and careful examination of what that existence is like and means. They believe that your work should not deny or push aside your experience as a being in the world. Instead, that experience should be connected to and affirmed in your work. In this way, they connect with cultural critics of science like Donna Haraway and cultural theorists like Gilles Deleuze and Félix Guattari, who stress the importance of personal experience as a component of disciplinary knowledge [Haraway, 1990b] [Deleuze and Guattari, 1987].

One of the tensions that has to be resolved in any work that combines science with non-scientific disciplines (of which Buddhism is certainly one!) is the differential valuation of objectivity. Generally speaking, the humanities tend to value subjective knowledge, whereas the sciences and engineering tend to prefer results that are objective. The notion of 'objectivity' is itself a can of worms, but we can work here with a preliminary understanding of objectivity as knowledge that is independent of anyone's individual, personal experiences. Since Varela, Thompson and Rosch want to connect cognitive science as science with individual human experience, they confront this problem of subjectivity versus objectivity head-on.

Interestingly, they do this by redefining what objectivity means with respect to subjective experiences. You cannot truly claim to be objective, they say, if you ignore your most obvious evidence of some phenomenon, i.e. your personal experience of it. This is particularly true when one is studying cognition —- in this frame of thought, any self-respecting study of the mind should be capable of addressing the experience of having one!

Given that one of the things cognitive scientists (and, by extension, AI researchers) are or should be interested in is subjective experience, Varela, Thompson, and Rosch abandon the focus on objectivity per se. But this does not lead to the long-feared nihilistic abandonment of any kind of judgments of knowledge — black is white, up is down, whatever I say goes, etc. Rather, they stress that Buddhist traditions have *disciplined* ways of thinking about that experience. The problem, they say, is not with subjectivity, but with being undisciplined. The goal, then, is being able to generate a kind of cognitive science that is subjective *without being arbitrary*.

Summary: Perspectives on Integrating AI and the Humanities

Generally, each of these researchers is interested in AI because of a fascination with the nature of human experience in the world. This interest naturally leads them to the humanities, which have dealt with questions of subjective human experience for hundreds of years. These researchers have found various ways to integrate this humanist experience with the science and engineering practices of AI. With respect to the issue of integrating AI and cultural studies that is pursued in this thesis, we can sum up their perspectives as follows:

- Winograd and Flores contrast existentialist philosophy with the analytic, rationalist philosophy that underlies much AI research. They use the differences between these approaches to understand the constraints that are inherent in AI methodology. They then develop new technology that, instead of being limited by these constraints, takes advantage of them.

- Suchman analyzes current AI practices to uncover the metaphors that underly them. These metaphors turn out to be specific to Western culture. She then asks what technology would be like if it were based on metaphors from a different culture.

- Chapman implements technology that is deeply informed by, among other things, the newly-identified metaphors of Suchman. He defends the concept that, though technology is well and good, fundamental *ideas* that are not testable in a scientific or mathematical sense are equally valuable to AI.

- Agre understands technical work as reflecting deep philosophical tensions. From this point of view, technical problems *are* philosophical problems. This means that the best progress can be made in AI by thinking simultaneously at the technical and at the philosophical levels.

- Varela, Thompson, and Rosch connect the science of human cognition with the subjective experience of human existence. They introduce, flesh out, and defend the idea that subjective does not necessarily mean arbitrary.

Each of these themes will be taken up in the work that follows.

Methodology: Subjective Technologies

Note to the technically trained: this section is philosophical and personal; its style may feel unfamiliar and uncomfortable for you. It inherits its style more from the traditions of cultural studies than technical work. I recommend trying to read it with a poetic rather than a technical frame of mind. If you do so, you may find that it not only lays out important foundations for the arguments that are to follow, but also betrays many secrets to the origins of my technical work that you, too, may find helpful in yours.
In this, you may find helpful the perspective of László Mérő: "My native language is rationality; my everyday logic cannot accept conclusions that contradict scientific results. Yet at the same time I clearly feel that there are many fields that slip out of the present range of science — and I do not deem them unworthy of reflection." ([Mérő, 1990],52)

The approach taken in this thesis follows Varela, Thompson, and Rosch in asserting that subjective experience, which goes to the heart of what it means to humans to be alive in the world, should be an important component of AI research. I believe that one of the major limitations of current AI research — the generation of agents that are smart, useful, profitable, but not convincingly alive — stems from the traditions AI inherits from science and engineering. These traditions tend to discount subjective experience as unreliable; the experience of consciousness, in this tradition, is an illusion overlaying the actual, purely mechanistic workings of our biological silicon. It seems to me no wonder that, if consciousness and the experience of being alive are left out of the methods of AI, the agents we build based on these methods come across as shallow, stimulus-response automatons.

In the reduction of subjective experience to mechanistic explanations, AI is by no means alone. AI is part of a broader set of Western cultural traditions, such as positivist psychiatry and scientific management, which tend to devalue deep, psychological, individual, and subjective explanations in favor of broad, shallow, general, and empirically verifiable models of the human. I do not deny that these theories have their use; but I fear that, if taken as the *only* model for truth, they leave out important parts of human experience that should not be neglected. I take this as a moral stance, but you do not need to accept this position to see and worry about the symptom of their neglect in AI: the development of agents that are debilitatingly handicapped by what could reasonably accurately, if metaphorically, be termed autism.

This belief that science should be understood as one knowledge tradition among others does not imply the rejection of science; it merely places science in the context of other, potentially — but not always actually — equally valid ways of knowing. In fact, many if not most scientists themselves understand that science cannot provide all the answers to questions that are important to human beings. This means that, as long as AI attempts to remain purely scientific, it may be leaving out things that are essential to being human.

In *Ways of Thinking: The Limits of Rational Thought and Artificial Intelligence*, for example, cognitive scientist László Mérő, while affirming his own scientific stance, comes to the disappointing conclusion that a scientific AI will inevitably fall short of true intelligence.

> In his book *Mental Models* Johnson-Laird says, 'Of course there may be aspects of spirituality, morality, and imagina-

> tion, that cannot be modeled in computer programs. But these faculties will remain forever inexplicable. Any scientific theory of the mind has to treat it as an automaton.' By that attitude science may turn a deaf ear to learning about a lot of interesting and existing things forever, but it cannot do otherwise: radically different reference systems cannot be mixed. (228-229)

But while the integration of science and the humanities (or art [Penny, 1997b] or theology [Foerst, 1998] [Foerst, 1996]) is by no means a straightforward affair, the work already undertaken in this direction by researchers in AI and other traditionally scientific disciplines suggests that Mérő's pessimism does not need to be warranted. We *do* have hope of creating a kind of AI that can mix these 'radically different reference systems' to create something like a 'subjectivist' craft tradition for technology. Such a practice can address subjective experience while simultaneously respecting its inheritances from scientific traditions. I term these perhaps heterogeneous ways of building technology that include and respect subjective experience 'subjective technologies.' This thesis is one example of a path to subjective technology, achieved through the synthesis of AI and cultural studies, but it is by no means the only possible one.

Because of the great differences between AI and cultural studies, it is inevitable that a synthesis of them will include things unfamiliar to each discipline, and leaves out things that each discipline values. In my approach to this synthesis, I have tried to select what is to be removed and what is to be retained by maintaining two basic principles, one from AI and one from cultural studies: (1) faith in the basic value of concrete technical implementation in complementing more philosophical work, including the belief that the constraints of implementation can reveal knowledge that is difficult to derive from abstract thought; (2) respect for the complexity and richness of human and animal existence in the world, which all of our limited, human ways of knowing, both rational and nonrational, both technical and intuitive, cannot exhaust.

"[T]he *interdisciplinarity* which is today held up as a prime value in research cannot be accomplished by the simple confrontation of specialist branches of knowledge. Interdisciplinarity is not the calm of an easy security: it begins *effectively*... when the solidarity of the old disciplines breaks down... in the interests of a new object and a new language, neither of which has a place in the field of the sciences that were to be brought together." ([Barthes, 1984], 169)

The Anti-Boxological Manifesto

The methodologies I use here inherit many aspects from the previous work described above. Following Winograd and Flores, I analyze the constraints that AI imposes upon itself through its use of analytic methodologies. Following Suchman, I uncover metaphors that inform current technology, and search for new metaphors that can fundamentally alter that technology. Following Chapman, I provide not just a particular technology of AI but a way of thinking about how AI can be done. Following Agre, I pursue technical and philosophical arguments as two sides of a single coin, finding that each side can inform and improve the other.

The additions I make to these approaches are based on a broad analysis of attempts to limit or circumscribe human experience. I believe that the major way in which AI and similar sciences unintentionally drain the human life out of their objects of study is through what agent researchers Petta and Trappl satirize as 'boxology:' the desire to understand phenomena in the world as tidy black boxes with limited interaction [Petta and Trappl, 1997]. In order to maintain the comfortable illusion that these black boxes sum up all that is important of experience, boxologists are forced to ignore or devalue whatever does not fall into the neat categories that are set up in their sstem. The result is a view of life that is attractively simple, but with glaring gaps, particularly in places where individual human experience contradicts the established wisdom the categories represent.

The predominant contribution to this tradition of humanistic AI which this thesis tries to make is the development of an approach to AI that is, at all levels, fundamentally anti-boxological. At each level, this is done through a contextualizing approach:

- *At the disciplinary level*, rather than observing a strict division of technical work and culture, I synthesize engineering approaches with cultural insights.
- *At the methodological level*, rather than designing an agent as an independent, autonomous being, I place it in the sociocultural context of its creators and the people who interact with it.
- *At the technical level*, rather than dividing agents up into more or less independent parts, I explicitly place the parts of the agent in relation to each other through the use of mediating transitions.

At all levels, my approach is based on this heuristic: "that there is no such thing as relatively independent spheres or circuits" ([Deleuze and Guattari, 1977], 4). My approach may feel unusual to technical workers because it is heavily metaphorical; I find metaphorical connections immensely helpful in casting unexpected light on technical problems. I therefore include in the mix anything that is helpful, integrating deep technical knowledge with metaphorical analysis, the reading of machines ([Mahoney, 1980]), hermeneutics, theory of narrative, philosophy of science, psychology, animation, medicine, critiques of industrialization, and, in the happy phrasing of Hayes and friends, "God knows what else." The goal is not to observe disciplinary boundaries — or to transgress them for the sake of it — but to bring together multiple perspectives that are pertinent to answering the question, "What are the limitations in the way AI currently understands human experience, and how can those limitations be addressed in new technology?"

Preview of Thesis Yet to Come

This phrasing of the fundamental question of the thesis may be a little too general for your tastes. In the next chapter, we will begin focusing on a detailed technical question: how to integrate many complex behaviors in an agent without degrading its overall quality of activity. The general goal of the thesis is to integrate engineering with humanistic perspectives; the concrete goal is to find technical solutions for behavioral degeneration by understanding its origin in the methodologies for agent interpretation and construction that are part of AI's scientific inheritance.

I will approach this goal in several steps. In Chapter 2, I will review current AI methodologies for synthesizing behavior, and uncover an inevitable limitation in its current approach. In Chapter 3, I will deepen this understanding by comparing AI approaches to agenthood with the methods of positivist psychiatry and scientific management; these black-boxing, objective approaches to human experience will be contrasted with contextualizing, subjective approaches critics of them have devised. In Intermezzo I, I will briefly introduce the "Industrial Graveyard," an implemented virtual environment that illustrates these objective and subjective approaches to agents, and forms the testbed for the technology developed in the thesis.

I use this notion of objective and subjective approaches to agents in order to develop a 'subjectivist' extension to alternative AI, termed socially situated AI, in Chapter 4. I use this approach to redefine the problems of behavioral disintegration in terms of audience *perception* of disintegration, and develop concrete technology to address it in Chapter 5. This involves the redefinition of behaviors as communicating *signifiers*, the development of *transitions* to synthesize behaviors, and the use of *meta-level controls* to implement transitions.

It turns out, however, that the approach of Chapter 5 is in practice too limited. Basically, it inherits an engineering perspective on the notion of audience perception that turns out to be inadequate in practice. In Intermezzo II, I take a brief detour into animation to find out how *animators* create the perception of authentically living beings. I combine this perspective with narrative psychology in Chapter 6 to form a new theory of intentional behavior based on the user's construction of narrative explanations. This 'narrative intentionality' forms the core of my developed agent architecture, the Expressivator, which is presented in its full glory in Chapter 7. With the cultural analysis and technical development of autonomous agents under our belt, Chapter 8 will return to the themes of the introduction, laying out how the work done here could form a part of a future integrated scientific-humanistic AI.

A Few Remarks on Format

This thesis is interdisciplinary between two fields that share little in their background knowledge or preferred rhetorical forms. Nevertheless, the work done here is not some AI work plus some cultural studies work; it is a single piece of work that has an AI face, a cultural studies face, and a large body in between.

The format of this thesis is intended to make comprehension of this undisciplined mass of knowledge as painless to the disciplinary reader as possible. The full body of the text is written in an attempt to be understandable to both the technically and the humanistically trained. However, the inclusion of all background knowledge that one or the other side may be missing would hopelessly balloon this thesis out of proportion and out of comprehensibility. When particular background knowledge is essential for just one discipline or the other to be able to make sense of the argument, that knowledge generally appears in sidebars to the text. Occasional sections (most notably the related work section in Chapter 5) lean heavily towards one side or the other. My hope is, however, that, for most of the thesis, no matter what your background, you will be able to negotiate a complete path through it, and find something in that path that is useful to you.

Chapter 2

Schizophrenia in Agents: Technical Background

The premise of this work is that there is something deeply missing from AI, or, more specifically, from the current dominant ways of building artificial agents. This uncomfortable intuition has been with me as an AI researcher for a long time, perhaps from the start, although for most of that time I was not able to articulate it clearly. Artificial agents seem to be lacking a primeval awareness, a coherence of action over time, something one might, for lack of a better metaphor, term 'soul.'

Roboticist Rodney Brooks expresses this worry eloquently:

> Perhaps it is the case that all the approaches to building intelligent systems are just completely off-base, and are doomed to fail. Why should we worry that this is so? Well, certainly it is the case that all biological systems.... [b]ehave in a way which just simply seems *life-like* in a way that our robots never do.
>
> Perhaps we have all missed some organizing principle of biological systems, or some general truth about them. Perhaps there is a way of looking at biological systems which will illuminate an inherent necessity in some aspect of the interactions of their parts that is completely missing from our artificial systems.... [P]erhaps at this point we simply do not *get it*, and... there is some fundamental change necessary in our thinking in order that we might build artificial systems that have the levels of intelligence, emotional interactions, long term stability and autonomy, and general robustness that we might expect of biological systems... [P]erhaps we are currently missing the *juice* of life. ([Brooks, 1997], 299-300)

This lack of 'aliveness' is not just a fuzzy intuition; it has its technical manifestations. One way in which this lack is expressed is in the difficulty of creating complex artificial creatures. A popular way of building these creatures in the alternative AI tradition is by composing behaviors. We have well-developed techniques for building behaviors which, by themselves, are clear, expressive and giving off the appearance of

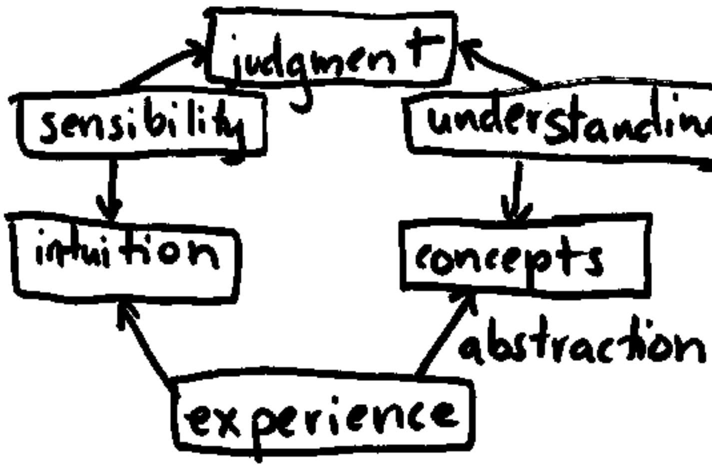


FIGURE 2.1: An agent structure inspired by Kant

life. The problem is that, as we try to combine more and more of these behaviors, the agent's overall activity gradually falls apart. If only a few behaviors are involved, the programmer can generally manage this disintegration. But when many behaviors are involved, their interactions are too complicated to be easily managed by hand.

The end effect of this difficulty is that, practically speaking, many complex behaviors simply cannot be adequately integrated. Instead, the agent tends to jump around from behavior to behavior, abruptly switching from one internally coherent behavior to another, its final activity a crazy quilt of actions with no coherent thread. These creatures, while perhaps intelligent in a formal sense, do not appear to have the coherence of behavior over time that we impute to living creatures. I term this overall incoherence *schizophrenia*, for reasons that will be thoroughly discussed later in this chapter.

In this chapter, we will examine this problem in the development of autonomous agents in detail. I give an overview of alternative approaches to agent construction, and then identify particular difficulties that tend to come up in synthesizing these agents. We will look at the construction of autonomous agents in depth to understand why schizophrenia happens. It turns out that the problem of schizophrenia is deeply connected with the way we think about building agents per se. Understanding this connection will provide the foundation for rethinking agent construction and addressing schizophrenia in the remainder of the thesis.

How to Build Yourself an Agent

It can sometimes be difficult for non-technical readers to imagine what exactly the parts of an agent might be, or how they could be connected to build a complete agent. Prior to delving into the guts of doing this from a technical point of view, I have taken the liberty of building two diagrams that show how an agent designed by a humanist might look (see Figures 2.1 and 2.2 — please take these with a liberal grain of salt). I

You should note that this way of conceptualizing humanistic traditions, while hopefully helping with the notion of agent construction, simultaneously does a grotesque violence to them, of a form which will become clearer in Chapter 3.

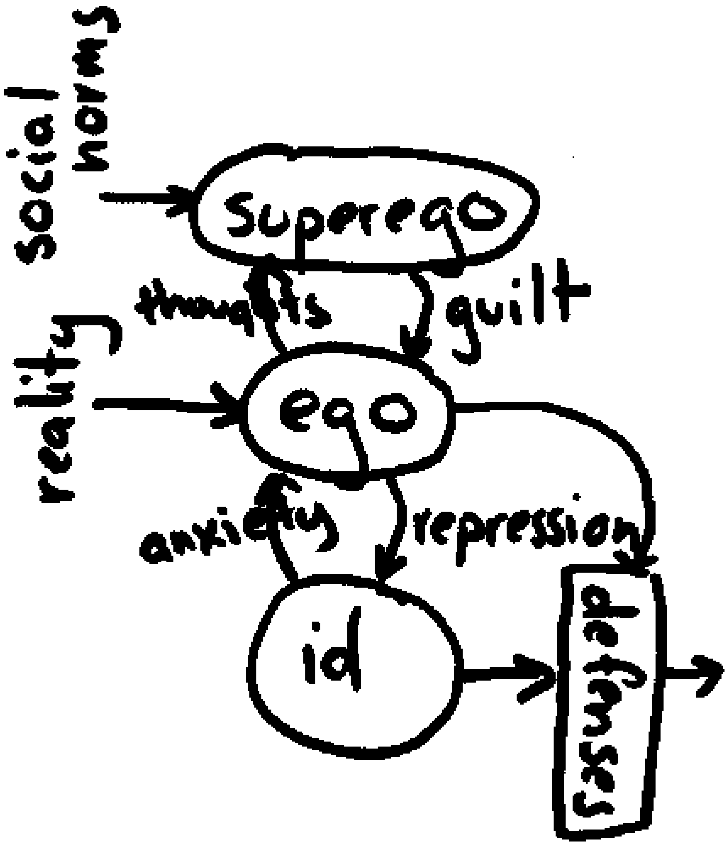


FIGURE 2.2: An agent structure inspired by Freud

heartily encourage the reader to design his or her own agent in the empty box provided for this purpose (Figure 2.3). Try to imagine what the various parts would be and how they might be interconnected. Please remember that anything not specified directly will not exist; do not exceed the boundaries of the box.

Pretty difficult, huh? My guess is that most people working from a humanistic tradition will quickly throw in the towel, since subjectivity is not something that can be simply diagrammed out on a piece of paper. AI researchers have no such luxury. The only way to build something is to specify it exactly. This means an essential part of agent construction is (a) deciding what the parts of an agent are and (b) deciding how the parts of an agent should be interconnected.

Until recently, the focus of the classical AI tradition has largely been on answering the first question. Through the mid-80's, classical AI research projects tended to focus on the development of isolated components for agents. Typically these components included natural language understanding systems, vision systems, memory modules, or planners. The final integration of agents into a complete, embodied, fully functional system was often deferred until the parts were sufficiently stable, which generally meant at some point in the distant future.

When some brave souls did attempt such integration, results were

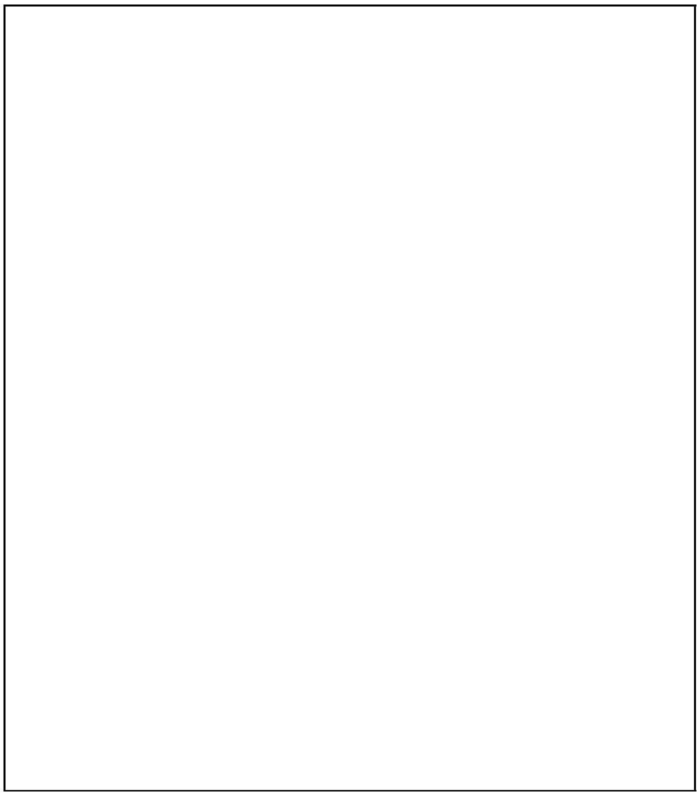


FIGURE 2.3: Draw your own agent here!

often disappointing, particularly in the robotic domain. Shakey [Nilsson, 1984], a beloved yet accurately named robot built in the late 60's and early 70's at the Stanford Research Institute, was one of the first attempts at building a complete agent. Shakey would go through cycles of sensing the environment, building an internal model of the outside world, deciding what to do, and doing it. Even in a carefully engineered environment, each of these cycles could take upwards of an hour. Splitting Shakey up into these sense-map-plan-act stages introduced computational bottlenecks that drastically affected its ability to react to a potentially changing environment.

More fundamentally, focusing on components and their subsequent specialization in research ghettos means that there are no forums to address their interrelationships. Systems are built with different logics, different input and output interfaces, and different assumptions about what the other systems will or can do. The temptation to leave out parts that are particularly difficult or ill-defined is strong, and there is no particular reason to resist it (or was, prior to the arguments of alternative AI). A somewhat crude but effective characterization of classical AI for humanists in this light is as the separate rationalization of part processes, with the eventual coordination of these processes into a coherent whole infinitely deferred.

Alternative AI defines itself in opposition to this approach as attempting to construct complete agents, from the ground up. The focus in these projects is often on building a complete agent first, then gradually improving its capabilities in a succession of more and more competent agents. A necessary and recurring preoccupation for these agent-builders, then, is the question of how the various pieces of an agent can be appropriately combined to form an at least semi-coherent agent.

Note that alternative AI does not have a monopoly on complete agent construction. A nice example of a classical agent is Homer [Vere and Bickmore, 1990], a virtual submarine that lives its own (rather dull) life under the sea, while taking orders from its buddies, Tim and Steve. It is also clear that many alternative AI projects also simply focus on small components. The point here is alternative AI's explicit commitment to and interest in integrated systems.

Typical agent parts:

Classical AI	Alternative AI
Perception	Object Avoidance
Modeling	Wandering
Planning	Wall Following
Execution	Picking Up Objects
Natural Language	Recharging Batteries

In the next sections, we will look at some of these projects in detail to identify alternative AI perspectives on integrated agent construction. I will focus both on the *concept* of agent used and on the agent *construction techniques*. Somewhat unsurprisingly, it turns out that these two aspects are inescapably intertwined.

As there are now enough proposed agent architectures to make several years of bed-time reading for an architecture junkie like myself, I have limited myself here to a smattering of architectures for which reasonably substantial agents have already been implemented. This is not intended to be a comprehensive coverage of behavior-based architectures, but to give a flavor of the type and range of architectures that fit under this umbrella term. In addition, in a perhaps vain attempt to not lose non-technical readers, I have kept the description of agent architectures following rather high-level, at the cost of doing some violence to the details of how each architecture works. For more general coverage, I suggest [Maes, 1990a], [Steels and Brooks, 1995], [Laird, 1991], or [Tyrell, 1993]. For more details on each architecture described here, please refer to the suggestions in each section.

Terminology

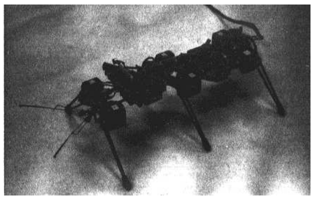
A few terms which are familiar to humanists in their colloquial sense will here be used in a technical sense. I will therefore briefly review the technical meanings of the most pertinent terms so that humanists are not immediately derailed.

- *Behavior* — A 'behavior' is a reified piece of activity in which an agent engages, for example 'sleep' or 'eat.' In colloquial English an agent behaves in various ways; in technical AIese, an agent has various behaviors.
- *The World* — When AI researchers speak of 'the world,' they mean the environment in which the agent is situated (not the Earth, for example). 'The world' is in contrast to 'the mind.'
- *Action* — An 'action' is an agent's most primitive unit of activity in the world. For typical artificial agents, actions will include things like picking up objects, rolling around, or moving arms and legs.
- *Function* — A 'function' is a reified ability which the agent has, which is often embodied in its own piece of code. Functions include things like being able to speak English, being able to see, or being able to reason about the consequences of actions.
- *Goal* — A 'goal' is a token which represents at a high level something which the agent is trying to achieve. Generally speaking, a goal is represented as a state of the world which the agent would

like to see happen: for example, that the car is parked without denting anything.

Subsumption Architecture

"Real biological systems are not rational agents that take inputs, compute logically, and produce outputs. They are a mess of many mechanisms working in various ways, out of which emerges the behavior that we observe and rationalize." ([Brooks, 1995], 52)



Brooks's Genghis

Rodney Brooks is one of the first, and certainly one of the most vocal, proponents of basing AI research on integrated agents from the start [Brooks, 1986a] [Brooks, 1986b] [Brooks, 1990] [Brooks, 1991b]. Brooks complains that previous approaches to building intelligence have often focused purely on building a "brain-in-a-box," i.e., defining agents only as information processors, without regard to their physicality. Automatic perception of the agent's physical environment, for example, has often been ignored, in favor of spoon-feeding agents human-designed descriptions of the world. Input and output being filtered through a human allows the researcher to showcase the intelligence of their subsystem, while avoiding the pesky little details of perception — which, it turns out, is extremely difficult.[1]

In contrast to this Cartesian, abstract subjectivity, Brooks sees agents as fundamentally physical and embodied. Rather than defining an agent in terms of abstract problem-solving — the chess-playing idiot savant – he thinks of it as behaving in a physical environment. The model for agenthood is inspired by biology and neurology ("Elephants don't play chess" [Brooks, 1990]), rather than human psychology. The prototypical Brooksian agent of the late 80's and early 90's[2] is the "Robot Insect." These insects are extremely limited in intelligence in comparison with traditional AI agents, but unlike these agents, they can walk rapidly around an office environment without killing anyone.

"True intelligence requires a vast repertoire of background capabilities, experience and knowledge (however these terms may be defined). Such a system can not be designed and built as a single amorphous lump. It must have components.... But true intelligence is such a complex thing that one can not expect the parts to be built separately, put together and have the whole thing work. We are in such a state of ignorance that it is unlikely we could make the right functional decomposition now. Instead we must develop a way of incrementally building intelligence." ([Brooks, 1986b],5)

Brooks's goal is to build complete agents that can function in a physical environment; he is less interested in the development of components than in the creation of complete agents, no matter how simple. As a consequence, he has problems with the way classical AI divides up its agents. He considers functional decomposition — the division of an agent into its hypothesized internal functions — to be an act of supreme intellectual arrogance. The claim is that we know so little about how agents are or should be constructed, that we will inevitably make bad choices and spend years of work on an extensive and well-designed module that will then simply be thrown away.

Since we have no way of knowing what the "proper" internal structure of an agent is, Brooks suggests that we should design an agent in terms of things we can see — its behavior. Each internally-defined agent behavior should directly connect perception of the world with action, causing humanly perceptible behavior. Just as evolution gradually builds up more and more complex animals, Brooks suggests creating more and more complex agents by adding new behaviors on top of old ones. The result is a hierarchy of behaviors, each of which is always active.

Brooks terms the typical classical AI method of dividing up agents 'horizontal decomposition' (Figure 2.4), because information from the

[1] ... though not un-tried, particularly by cyberneticists.

[2] More recently, Brooks has been building a humanoid robot that models early infant development [Brooks and Stein, 1993].

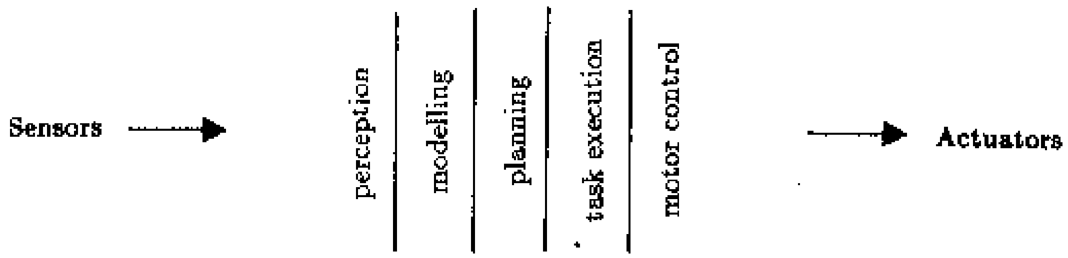


FIGURE 2.4: Horizontal Decomposition

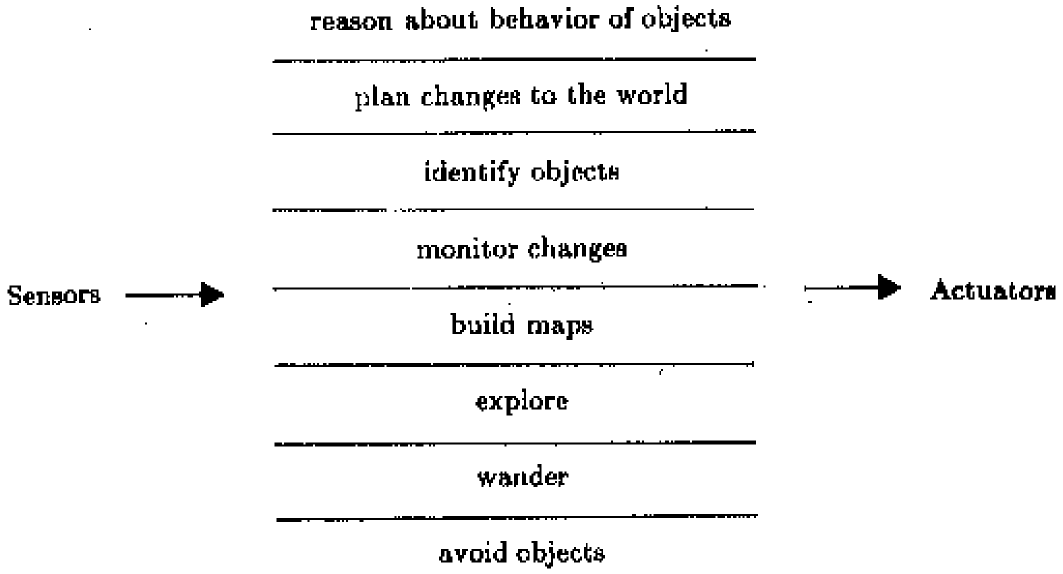


FIGURE 2.5: Vertical Decomposition

environment needs to flow through all the parts of the agent before affecting its externally observable behavior. His own system he terms 'vertical decomposition' (Figure 2.5), in which every designed module forms a direct link between external environmental input and observable behavior. In this system, the 'parts' of an agent are behaviors, each of which connects perception to action, i.e. whose effects are directly observable by its builder.

Behaviors, in this scheme, are built separately. Behaviors are not aware of each other; each is designed as a self-contained unit. Communication between behaviors is possible, though limited, but most communication occurs by observing the results of other behaviors' actions in the world. This means behaviors are very loosely coordinated. Behaviors are thought of as many self-contained parts, only locally interacting, an idea Brooks and many other alternativists inherit from Marvin Minsky [Minsky, 1988].

Behaviors are combined through a process of layering, wherein all behaviors function simultaneously. Higher-level behaviors can subsume lower-level behaviors by blocking their output, or by providing them with false input. Behaviors that do not subsume each other can influence one another using a 'hormonal' system, inspired by Maes (page 33) which provides a kind of global state [Brooks, 1991a]. Behaviors can release 'hormones' which then may trigger other behaviors to be active. Often times, conflicts between behaviors are avoided by having them only be active under particular conditions, so that they are not likely to engage in action at the same time.

Subsumption Architecture Agent Design Strategy

1. Decide what the agent should do.
2. Decompose this into behaviors in a hierarchy from simple to complex.
3. Start building behaviors from the bottom up, starting with simplest.
4. Once the simplest behavior works, design the next behavior on top of that.
5. Continue until all behaviors function.

Pengi

"The world of everyday life... is not a problem or a series of problems. Acting in the world is an ongoing process conducted in an evolving web of opportunities to engage in various activities and contingencies that arise in the course of doing so. Most of what you do you already know how to do, and most of the rest you work out as you go along. The futility of trying to control the world is, we think, reflected in the growing complexity of plan executives. Perhaps it is better to view an agent as *participating* in the flow of events. An embodied agent must *lead a life*, not *solve problems*." ([Agre and Chapman, 1990], 20)

One of the vital subfields of AI during the last 20 to 30 years is 'planning,' i.e. the selection of actions by an artificial agent in order to achieve its goals in the world. Prior to the mid-80's, planning algorithms typically had 3 parts: perception, plan-building, and execution. The perception phase (often short-circuited by the spoon-feeding methodology mentioned above) was used to build an internal model of the outside world. Plan-building took up the bulk of the effort, and generally consisted of mentally trying out all possible actions in the model of the world to try to find a sequence of actions that would cause the given goal to be achieved. Execution came after the fact and consisted of actually doing each step in the decided-on plan. Assuming that the planner was able to take into account every contingency, and that the executor could accurately do the actions given to it, this worked correctly even for complex goals.

This approach to agent construction places most of the burden of agent activity on reasoning about and manipulating a model of the world. Most of the agent-building effort is spent on thinking about the world, and very little on perceiving and acting. A lot of effort goes into considering contingencies and expecting the worst from a hostile environment. In the mid-80's, Phil Agre and David Chapman developed an agent, Pengi, based on a radically different model of agenthood [Agre and Chapman, 1990].

Agre and Chapman understand agents not as thinkers in a hostile world, but as doers situated in a usually benign environment. The agent spends most of its time in routine behavior, not in the planning out of details of action. Most of the agent's behavior is more or less automatic; variation and improvization happen as the agent responds routinely to a changing environment, rather than from the agent's flexibility in deciding complex sequences of actions.

Fundamentally, Agre and Chapman base their agent structure on the belief, heavily influenced by Lucy Suchman's description of situated action [Suchman, 1987], that intelligence should be understood in terms of interaction between an agent and environment, rather than in terms of the manipulations of an agent of a hostile world. For them, the challenge is to understand how routine behavior can arise and adapt to a changing environment, rather than how the system can anticipate and plan for every possible contingency.

Pengi

Given this interaction-oriented outlook, the separation of perception, planning, and execution no longer make sense. All parts of the agent should be integrated and tightly coupled with sensing and action. For Agre and Chapman, the parts of the agent are based on simple routines in which the agent should engage when placed in a particular environment. These routines are decomposed into actions, with the rationale for each action analyzed. The rationale for actions is then reduced to conditions in the environment that the agent can sense. An agent, then, consists of physical actions that are cued by sensed conditions.

In order to maintain tight coupling with the environment, agents no longer engage in a long-term perceive - think - act cycle. Rather, at every time step the agent must choose an action to take immediately based on conditions in the world. The routines the designer chose may or may not happen, since the actions are continuously redecided and in the middle of executing one "plan" actions from other plans may make more sense. The problem of actions conflicting is avoided by specifying enough conditions for each that there is only one 'right' action. It's not totally clear how this solution would work for an agent with many high-level routines, not all of which can be decided based on perceivable things in the world (Maes's architecture, described next, is in part a reaction to this).

Agre and Chapman Agent Design Strategy

1. Examine the agent or desired activity to find typical 'routines' one would engage in (often using ethnographic techniques).
2. Decompose these routines into actions. Determine rationale for each action.
3. For each action, find conditions in the world that should trigger that action according to its rationale.
4. For actions that are triggered at the same time, find additional conditions to let you choose between them.

Agent Network Architecture

Agre and Chapman's architecture has the advantages of being adaptive and reactive to changes in the environment. Pengi is an improviser who sometimes makes mistakes, but can go with the flow to generally come out on top. Pengi is fundamentally the reflection of a theory of human action, and is not intended as the peak of technological competence. You might like Pengi very much, but you probably don't want it to be running the US nuclear warhead control system.

"Given an agent that has multiple time-varying goals, a repertoire of actions that can be performed..., and specific sensor data, what actions should this agent take next so as to optimize the achievement of its goals?" ([Maes, 1993 1994],146)

For Pattie Maes, the functionality of agents is more important than theories of human agenthood. While the reactivity that comes from situated approaches is important, she is not wedded to Chapman and Agre's idea that agents are or should be fundamentally improvisers. Maes's agent definition is basically technical and functional, rather than psychological or biological; her examples of agents include planetary explorers, shop schedulers, and autonomous vacuum cleaners. As a consequence, for Maes the important thing is getting the agent to have the proper functionality. She defines the fundamental problem of agenthood as *action-selection*:

> Given an autonomous agent which has a number of general goals and which is faced with a particular situation at a specific moment in time. How can this agent select an action such that global rational behavior results? [Maes, 1989a]

Appropriately enough, her architecture, the Agent Network Architecture (ANA), is often nicknamed "Do the Right Thing."

With this outlook, Maes still uses a behavior-based approach, but takes a different point of view on the question of how behaviors should be integrated. It is very unlikely that a designer will be able to foresee all possible combinations of events in the environment so that the agent will always take the right action. Instead, she wants to let her agents do some reasoning to figure out the best action to take, though she does not want to return to a system where reasoning dominates over action in the environment. In order to do this, she has developed a sophisticated action arbitration mechanism to let the agent quickly and mostly correctly decide which action it should take.

Maes divides her agents into "competence modules," which basically correspond to behaviors for Brooks [Maes, 1990c]. A competence module is capable of taking some kind of action in the world, related to the tasks for which the agent is programmed. Competence modules are grouped according to how they relate to the overall goals of the agent. Competence modules basically act on their own, but they allow for low-bandwidth communication to decide which module should be active. All competence modules are always active, but they are only allowed to actually do something if they are activated using a spreading activation system.

Humanists, Maes's technique of spreading activation is related to the hormonal system described on p. 32. It is loosely based on analysis of neural networks.

Specifically, modules are connected to each other according to the logic of their organization for a task. To put it a little too simply, modules have positive links with other modules that make them possible; and they have negative links with other modules that make them impossible. To start out, modules get "energy" if they are possible in the world, or if they are desired goals. Modules then spread energy over the positive links, and block energy over the negative links. The result is that, on average, the module that is most likely to help achieve the most important goal is chosen. [3]

Agent Network Architecture Agent Design Strategy

1. Choose a set of goals for the agent in its environment.
2. Identify tasks that will allow the agent to achieve each of the goals.
3. Break each task into its component actions.
4. Determine the preconditions and effects of each action
5. Determine how actions affect each other: which actions make other actions possible, which actions undo the work of other actions
6. Make links between actions according to how they affect each other

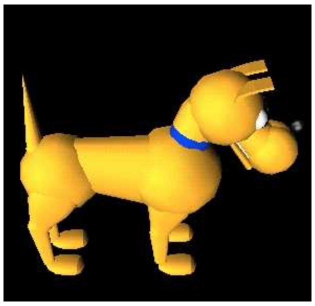
[3] This explanation is of necessity extremely simplistic. I apologize and refer interested readers to [Maes, 1989b].

Hamsterdam

Bruce Blumberg builds on ANA by taking very seriously the notion of agent as animal [Blumberg, 1994]. Ethology — the study of animal behavior — has been an at least vague source of inspiration for many alternative AI researchers, but Blumberg integrates ethological principles into agent architecture to a new degree. His system, Hamsterdam, can be seen as a hybrid of the Maes goal-achieving optimality approach with ethological principles, based on the question: how can a creature, whether biological or artificial, decide, at each point in time, what is best for it to do?

Blumberg extensively adapts concepts from ethology in order to be able to build artificial creatures that share some of the properties ethology has identified as belonging to living creatures. For Blumberg, then, the 'units' of his agents are behaviors, as understood by ethologists. This means black boxes like "walk" or "sleep," with only simple interaction between them. Behaviors are related to drives or needs (hunger, fatigue) which they can fulfill. Behaviors are hierarchically organized into "behavior groups," which represent alternative ways to fulfill the same drive.

"[W]e wish to build interactive 'life-like' creatures such as virtual dogs, beavers, monkeys, and perhaps one day characters which are caricatures of humans. The fundamental problem for these creatures (as well as for their real world counterparts) is to 'decide' what among its repertoire of possible behaviors it should perform at any given instant." ([Blumberg, 1996], 29)



Blumberg's Silas

Blumberg's technique of combining behaviors is based on action-selection. The agent continuously redecides its actions, so that at any point in time the creature is engaging in the 'best' behavior (where 'best' is a combination of environmental appropriateness with factors such as maintaining a persistent focus of attention). Behaviors compete for control of the body, using a 'winner-take-all' scheme that works as follows: Behaviors constantly monitor the environment for conditions under which they might be appropriate. When they are triggered, they calculate a value that represents their appropriateness. Roughly speaking, the behavior with the highest value is allowed to take an action; 'losing' behaviors may suggest actions which the 'winning' behavior may or may not also take (for more details see [Blumberg, 1996]).

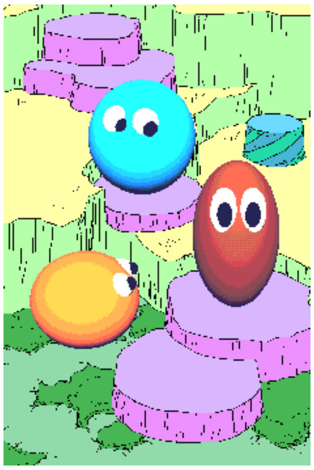
Hamsterdam Agent Design Strategy

1. Choose a creature in the world or a character to model
2. Decide on the needs and drives of the creature
3. Decide what behaviors the creature has, and how they fulfill the chosen needs and drives
4. Cluster related behaviors together into groups according to how they contribute towards the agent's actions
5. Manipulate the variables used to select behaviors in each group to get appropriate behavior under different circumstances
6. Manipulate the variables used to select behaviors between groups to get appropriate overall behavior in different circumstances

Hap

Loyall and Bates's Hap [Loyall and Bates, 1991] [Loyall, 1997a], the system on which this thesis work is based, is similar in many respects to Hamsterdam and a number of other reactive architectures. It is, however, the first such agent architecture to be focused on agents which are characters, rather than agents as animals or tools. Essential to the Hap understanding of agents is that an artist is attempting to express their vision of a particular character through the constructed agent. Conceptually, the Hap agent, while generally not amazingly intelligent, is "believable" as a character. This means it conveys a strong, author-chosen personality while avoiding doing anything so wrong that its audience is jarred out of belief in the agent as a living being.

" 'Believable' is used here in the sense of believable characters in the arts, meaning that a viewer or user can suspend their disbelief and feel that the character or agent is real. This does not mean that the agent must be realistic. In fact, the best path to believability almost always involves careful, artistically inspired abstraction, retaining only those aspects of the agent that are essential to express its personality and its role in the work of which it is a part." [Loyall, 1997b]


The Woggles

For Hap, it is not so important that the creature do the right thing with respect to fulfilling goals and drives in the environment. Rather, it is important that the agent be able to express its personality clearly down to the details of its behavior. At the same time, the agent must clearly react to what happens around it, appear to engage in goal-oriented behavior, be aware of what other characters and human interactors are doing, and in general not do anything that breaks the audience's suspension of disbelief. This means that the Hap architecture needs to combine the reactivity and environment-centeredness of other alternative AI architectures with a greater emphasis on author control of the details of behavior, rather than having behaviors be more or less generic, or having the details of the behaviors gradually emerge from what the agent decides to do.

The Hap architecture splits agents into goals and behaviors. Goals are simply names that represent to authors what the agent is doing (e.g. "dance").[4] Behaviors are intended as methods for doing goals, and they consist of author-written collections of physical actions (e.g. "jump") and other goals. Behaviors are made reactive by annotating them with environmental conditions under which they are or aren't appropriate to do; a behavior that is running will terminate itself when and if it becomes inappropriate.

When behaviors run, they can simultaneously start multiple goals. After some time, then, an agent may be pursuing quite a few goals simultaneously. Interaction between goals is handled by a priority mechanism, in which goals of high priority will be chosen over goals of lower priority. In addition, the author can mark particular combinations of goals as conflicting, so they can never happen at the same time. Additional details on Hap can be found in [Loyall, 1997a].

Hap Agent Design Strategy

1. Design a character to be implemented, including typical behavior and personality
2. Choose a set of high-level goals the character will engage in
3. For each goal, write a set of behaviors that instantiate that goal in different situations in a way appropriate to the character's personality
4. Each behavior may introduce new goals, so continue step 3 until all goals have behaviors
5. Add annotations to goals that conflict with each other

Summary: Alternative AI Agent-Building

Each of the listed architectures adds something important to the mix that is alternative AI. For the sake of the argument here, the following aspects are most important:

[4] Note this is different from the definition of 'Goals' given earlier.

- From Brooks comes the concept that agents should be divided into behaviors, each of which can run independently. Behaviors are basically independent, though there may be some low-bandwidth communication between them.

- Chapman and Agre introduce the idea that if an agent is to be situated responsively in an environment, it should redecide its behavior on every time step. By continuously redeciding behavior, the agent immediately responds to changing environmental conditions. These rapid alterations in behavior lead to the generativity of their architecture.

- Maes makes the critique that, for reasonably complex agents, decisions about how to arbitrate between behaviors cannot be made ahead of time. Therefore, agents will need to be able do some reasoning on their own. Maes introduces and Blumberg refines the concept of action-selection, i.e. that at every time step the agent should choose an action that is 'best' according to its goals or drives.

- Loyall and Bates add the concept that an agent should be written with an eye to how it affects its audience.

These architectures have disparate views of what an agent is, taken from different backgrounds: biology, ethnomethodology, engineering, ethology, and character animation. At the same time, a generally shared picture of agent construction emerges:

Note these are not the only commonalities between these architectures, or the only characteristics defining alternative AI. These are simply the ones most pertinent at this stage of the argument.

- Agents are seen as situated in an environment. Therefore, an agent's 'parts' are behaviors, which each may result in visible action in the world. Each behavior is firmly anchored to perception of the environment (when am I appropriate?) and to action upon the environment (what should I do?).

- Behaviors run relatively independently of one another. Each behavior does its own sensing, world modeling (where necessary), and makes its own decisions about appropriate action. Behavioral coordination and communication is minimal. All behaviors are running all the time, or at least when they are possibly appropriate. An agent may or may not actually simultaneously take actions caused by multiple behaviors.

- Conflicts between behaviors are handled with respect to what is most appropriate under given environmental conditions, and, for some architectures, with respect to what is most appropriate given current goals, emotions, drives, and / or recent actions.

- In order to remain reactive, agents continuously redecide their behavior in light of changes in the environment (as well as changes to their internal state).

Schizophrenia as a technical problem

One of the fundamental complaints alternative AI makes about classical AI is that it focuses on the functional components of intelligence. These components are generally hard to integrate into a complete agent. Their underintegration can manifest itself, for example, in various kinds of inconsistency between the different functions, such as not being able to use knowledge for one function that is available for another. So the agent may speak a word it cannot understand or visibly register aspects of the world that do not affect its subsequent behavior.

In contrast to this functional decomposition, alternative AI designs behaviors, each of which integrate the functionalities it needs to operate. This does not solve all problems, however, since the various behaviors also need to be integrated. Brooks, for example, has stated that one of the challenges of the field is to find a way to build an agent that can integrate many behaviors, where he defines many to be more than a dozen [Brooks, 1990]. In complex agents that exhibit many behaviors, those behaviors are extremely difficult and tedious to integrate completely, with the result being that they often remain only loosely integrated.

The reason for this difficulty can be traced to a fundamental tenet of the behavior-based approach. The design choice in behavior-based AI is to build behaviors independently, and have them use minimal communication and coordination. This black-boxing approach has the advantage of simplifying agent design, since each behavior can be designed and built separately. It can also give you a complete, though limited, agent sooner, since each behavior is in effect a complete agent. Nevertheless, the black-boxing approach raises the question of how the different behaviors of the agent can be made to work together properly. The next section gives a concrete example of these problems; this will put us in position to formally define the difficulties of integration for alternative agents.

Case Study: Integrating the Woggles

In 1992, a group of 13 researchers, including Oz Project members, built "The Edge of Intention" [Loyall and Bates, 1993] (Figure 2.6, a system containing small, social, emotional agents called "Woggles" that interact with each other and with the user. We used the Hap architecture to build these agents.

Following the Hap design strategy, we first built a set of high-level behaviors such as sleeping, dancing, playing follow-the-leader, moping, and fighting. Each of these behaviors was reasonably straightforward to implement, including substantial variation, emotional expression, and social interaction. Watching the agents engage in any of these behaviors was a pleasure.

Then came the fatal moment when the behaviors were to be combined into the complete agent. This was a nightmare. Just combining the behaviors in the straightforward way led to all kinds of problems:

- Agents would try to engage in two behaviors simultaneously that did not make sense (e.g. , 'fight' and 'sleep' — we optimistically called the result "emergent nightmares").
- Agents would switch from one behavior to another with their body in an unusual state. For example, an agent startled out of sleeping might engage in several subsequent behaviors with its eyes shut.

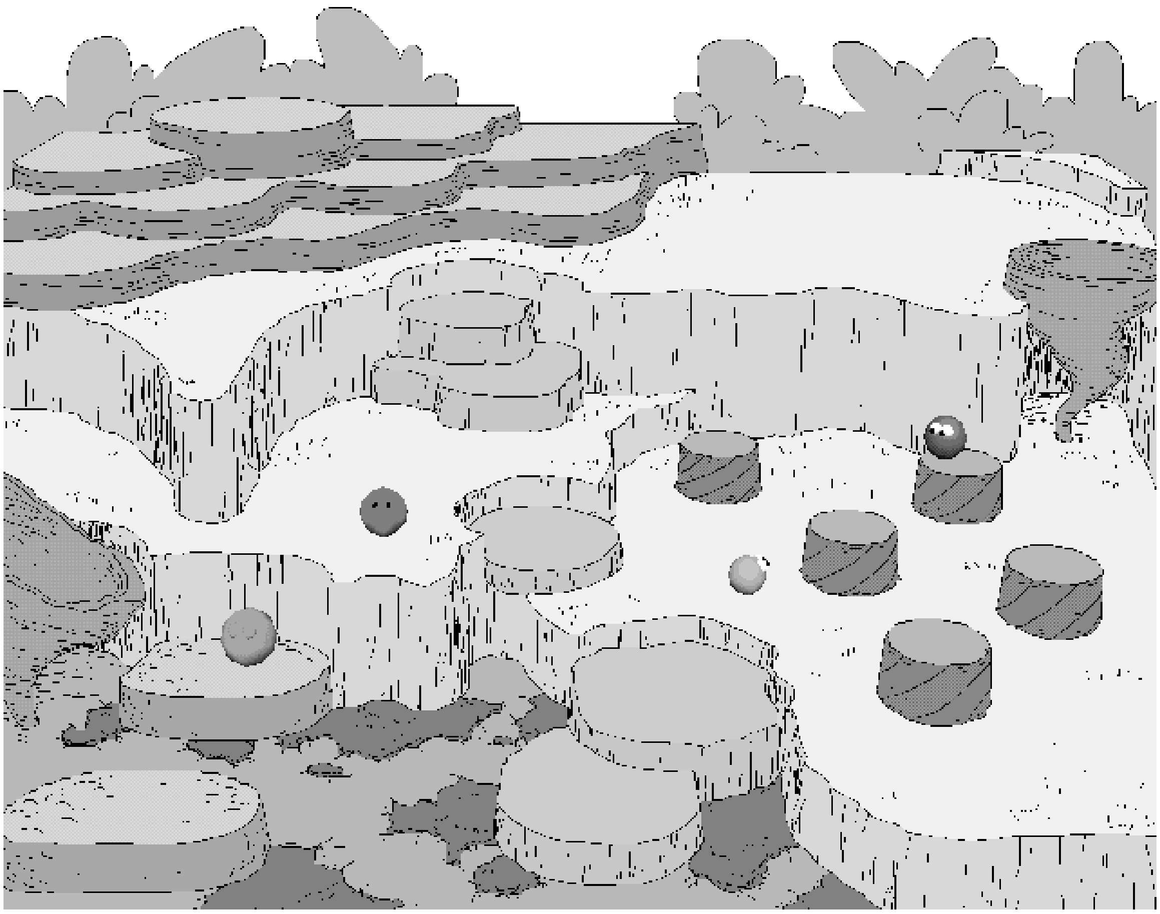


FIGURE 2.6: The Edge of Intention

- Agents would rapidly switch from behavior to behavior, never settling into one long enough to make the resultant activity comprehensible. Alternatively, agents would refuse to switch from one behavior to another in situations where they really should change, making it seem that the agent was clueless about what it was doing.
- Agents would get stuck in 'loops' where they kept switching back and forth between two behaviors, never being able to settle down into one until something in the environment drastically changed.

Under the pressure of deadlines, we added an ad hoc system to handle interbehavioral coordination: agents could only engage in one high-level behavior at a time; when switching from behavior to behavior, we reset the most crucial aspects of the body (open the eyes, stop trembling, stand up straight, etc.); express personality by varying the probability that you would engage in a particular high-level behavior under circumstances where it is appropriate. This clearly improved matters, but it did not fundamentally solve any of the problems, and they still regularly reared their ugly heads during runs of the system.[5] While individual behaviors

[5]Loyall believes that many of these problems were rooted in a bug in the way in which

were easy to write, the interactions between behaviors — particularly as manifest in the agent's apparent activity — were difficult to control and manage properly. These problems are not unique to Hap.

Schizophrenia defined

Alternative AI, while clearly having some impressive results, has not solved a fundamental difficulty of classical AI, i.e. its inability to integrate the parts of the agent into a coherent and coordinated whole. Generally, the agent's behaviors are too crystallized; the boundaries between the agent's behaviors are too sharp. Unlike biological agents (or characters in a film, for that matter), one can see the boundaries between the agent's behaviors.

In particular, alternative AI agents generally have a set of black-boxed behaviors. Following the action-selection paradigm, agents continuously redecide which behavior is most appropriate. As a consequence, they tend to jump around from behavior to behavior according to which one is currently the best.[6] What this means is that the overall character of behavior of the agent ends up being somewhat deficient; generally speaking, its behavior consists of short dalliances in individual, shallow high-level behaviors with abrupt changes between behaviors. It is this overall defective nature of agent behavior, caused by under-integration of behavioral units, that I term *schizophrenia*.

Because all behavior-based systems do not integrate behaviors in the same way, they also do not exhibit schizophrenia in the same way. Some of the difficulties with Hap are noted above. Each of the other architectures has its own style of schizophrenia, which is best observed visually or through the experience of programming, but can sometimes be gleaned from research reports.

- Brooks's experience seems to parallel ours with the woggles. Adding new low-level behaviors to his robots is straightforward using the subsumption technique. However, Brooks does not even try to integrate many high-level behaviors; he states up front that it is not currently possible. Getting coherent overall behavior is an open question: "A humanoid robot has many different subsystems, and many different low level reflexes and behavioral patterns. How all these should be orchestrated, especially without a centralized controller, into some sort of coherent behavior will become a central problem" ([Brooks, 1997], 297).

- Pengi jumps from action to action according to whatever seems most appropriate from moment to moment. As a consequence, Pengi mixes its behaviors together in ways that may or may not result in activity that seems to make sense. As Agre and Chapman charmingly put it, "Pengi regularly... combines its repertoire of activities in useful ways we didn't anticipate. (It also regularly does silly things in situations for which we haven't yet wired it)" ([Agre and Chapman, 1990], 23).

conflicts between goals were handled [Loyall,]. Subsequent experience by other designers with a debugged version of the system ([Neal Reilly, 1996], Chapter 7 of this document) suggests that while this may have been part of the story, substantial problems remain.

[6] A similar observation is made by Steels [Steels, 1994].

- Like Pengi, ANA's schizophrenia manifests itself in jumps from action to action. Nevertheless, the silliness of the results is mitigated somewhat by the fact that the system itself is doing some reasoning about what is appropriate to do. It is a little difficult to judge it accurately, though, without being able to see the dynamics of the system in action, preferably on a set of complex high-level behaviors.

  Some provisional conclusions may be drawn from Bradley Rhodes's PHISH-Nets [Rhodes, 1996], which built on ANA, and was used to implement several characters. These characters displayed a kind of schizophrenia where they could reason extensively about conditions in the environment, but then moved abruptly and rather woodenly from atomic behavior to atomic behavior. While this may be more an indicator of the limits of a master's thesis system than an inherent characteristic of ANA, it seems likely that, if used to drive a graphically represented agent, ANA would have the tendency, like Hap, to switch rapidly from behavior to behavior, and to get stuck in behavioral loops.

- Silas, the dog built using Hamsterdam, is like Pengi in jumping from behavior to behavior. Unlike Pengi, Silas's individual behaviors are well-integrated, so it is fairly clear which behavior Silas is engaging in. Unfortunately, this increase in behavioral coherency also increases Silas's apparent schizophrenia, since it leaps from behavior to behavior, in a way that is clear and can be abrupt and jarring. Often, there is no clear thread connecting the behaviors, resulting in an appearance of either behavioral randomness or pure stimulus-response.

While schizophrenia manifests itself in different ways, it can generally be understood as a manifestation of the limit point of behavior integration. Programmers can create robust, subtle, effective, and expressive behaviors, but the agent's overall behavior tends to gradually fall apart as more and more behaviors are combined. For small numbers of behaviors, this disintegration can be managed by the programmer, but as more and more behaviors are combined their interactions become so complex that they become at least time-consuming and at worst impossible to manage. Schizophrenia is the symptomatology by which behavioral underintegration can be directly observed in the agent. It manifests itself in at least two ways that make the resulting system hard to understand: (1) switching abruptly and mechanically from high-level behavior to high-level behavior; (2) mixing actions from different behaviors together in an incoherent jumble .

Why schizophrenia?

At this point, you may be wondering to yourself why on earth I am using the psychiatric term 'schizophrenia' for this technical difficulty. If so, good for you! Schizophrenia is a complex term, loaded with a history of contradictory uses and abuses in a variety of fields, and so full of 90's cultural theory cachet that observers may wonder if it really still means anything at all.

It is with some trepidation, then, that I introduce this term now in a technical context. I believe that the reasons for its use in this case are so compelling that they outweigh the dangers of adding to the obfuscation that already exists around this term. In particular, many uses of the term schizophrenia bear deep relations with the problem of behavioral underintegration in alternative AI. Giving these usages the same name allows for the development of their metaphorical connections, making it potentially illuminating to look at all these versions of schizophrenia simultaneously. In this respect, focusing on this technical problem as a variation on schizophrenia may actually increase understanding of schizophrenia, rather than further diluting the term.

While an extensive examination of schizophrenia will have to wait until Chapter 3 and Intermezzo I, I will here explain how schizophrenia has historically been used in psychiatry and cultural theory, and clarify how it relates to current problems in AI. The key point will be the multiple uses of schizophrenia as a metaphorical concept, and how they each put the difficulties of alternative AI in a new light.

1.Schizophrenia as incoherence

The notion of schizophrenia as a psychiatric term is generally seen as originating with Kraepelin, who unified a variety of disorders under the name *dementia praecox* in 1898. This name was intended to refer to the fact that these disorders all seemed to be related to a gradual mental deterioration that began when the patient was young. In 1911, Bleuler renamed this heterogeneous group of disorders *schizophrenia* "because he thought the disorder was characterized primarily by disorganization of thought processes, a lack of coherence between thought and emotion, and an inward orientation away from reality. The 'splitting' thus does not imply multiple personalities but a splitting within the intellect and between the intellect and emotion" ([Coleman *et al.*, 1984], 344). The usages of the term schizophrenia have tended to cluster around the description which Bleuler gave, specifically emphasizing an internal incoherence and disorganization. The incoherence we see in alternative AI agents, then, can be put in a broader light: it corresponds at a high level with some conceptions of schizophrenia from psychiatry. This will be the most basic, and most inaccurate, usage of schizophrenia here.

2.Schizophrenia as a meta-level incoherence

Schizophrenia has never been a straightforward, easily identifiable syndrome. The heterogeneity of the disorders and symptomatology to which the term schizophrenia can be applied has led to a substantial amount of diagnostic creep in this "most baffling" ([Coleman *et al.*, 1984], 345) of psychiatric disorders, including substantial variation between geographical regions and over time. The Diagnostic and Statistical Manual of Mental Disorders (DSM) [American Psychiatric Association, 1980], the official repository of definitions of mental illness, has reflected these variations.

> [T]he criteria are a curious mixture of an older set of concepts originally proposed by Bleuler (1911, 1950) and a newer set, chiefly those of Schneider (1959), which appear to have only an obscure and unspecified relationship to each other. In consequence, we cannot be sure that persons on whom much of our research knowledge depends — those

> who were diagnosed as schizophrenics under, say, DSM-II — can be grouped with persons described as schizophrenic under DSM-III. In one study peripherally concerned with this dilemma, a group of 68 DSM-II-defined schizophrenic patients was reduced to 35 when DSM-III criteria were applied, a reduction of 51 percent! ([Coleman *et al.*, 1984], 353)

Even using one particular criterion, the concept of schizophrenia is hard to pin down, with proliferating subcategories, symptoms, and relations, rather than a set of properties with a common core. "There is, in fact, no constant, single, universally accepted 'sign' of the presence of schizophrenia;" ([Coleman *et al.*, 1984], 354) this psychological textbook concludes that the only common feature is behavior that is bizarre and unintelligible ([Coleman *et al.*, 1984], 353).

What's interesting, then, is that schizophrenia refers to a kind of incoherence, but is itself incoherent as a concept. It is notoriously difficult to pin schizophrenia down as a particular thing, a fact which reflects itself in the multiple metaphorical uses I list here. It is equally difficult to classify people accurately and repeatedly as schizophrenic. Schizophrenia in psychiatry, then, can also be understood as a meta-level problem: the difficulty of understanding and classifying people within a rational system. Schizophrenia in this sense represents the limits of our ability to categorize people.

Categorization enters into alternative AI as well: the first step of designing an agent requires us to divide the agent's overall, perhaps ineffable behavior and personality into a set of relatively clearly-defined behaviors. Schizophrenia as meta-level incoherence suggests that this step is fraught with danger, since there may be limits to our ability to understand and categorize behavior and those limits may manifest themselves in incoherence at the level of synthesis. The concrete implications of this for alternative agents will become more apparent in the analysis of agent construction later in this chapter.

3.Schizophrenia as a theory of consciousness

As noted in Chapter 1, schizophrenia for cultural theorists refers to a particular way of thinking about what it means to be human in contemporary Western society. This usage came about in response to perceived difficulties with the rational model of subjectivity. This is because the rational model no longer works when we talk about people who have traditionally been marginalized. If the rational is the definition of what it is to be human, it is equally true that disenfranchised people, such as women and blacks, have often been classified as nonrational and hence as unworthy of the status of full humans. For example, when we deal with the mentally ill, we are dealing with people who by definition are nonrational [Foucault, 1973].

The use of the term 'schizophrenia' to describe a kind of subjectivity that could apply to everyone — not just the mentally ill — is inspired by the antipsychiatric movement of the 1960's. The antipsychiatrists seek to include those with mental illnesses in the category of the 'human' by describing their mental processes as simply more extreme versions of processes that take place in everyone's mind, rather than as the fundamentally different (nonrational) way of thinking the rational model has to assume. In short, a schizophrenic model of consciousness takes into account the experiences of (for instance) the mentally ill in order to create a more inclusive model of human experience.

"[W]e believe that abstract reasoning is *perforated*: it is not a coherent module that systematically accounts for all or even a class of mental phenomena. It is not a general-purpose reasoning machine, as it appears to be, but only a patchwork of special cases." ([Chapman and Agre, 1986] 415)

"The existential concern that animates our entire discussion in this book results from the tangible demonstration within cognitive science that the self or cognizing subject is fundamentally fragmented, divided, or nonunified." ([Varela *et al.*, 1991], xvii)

This antipsychiatric belief that normal human existence is fundamentally irrational and incoherent, with a thin veneer of apparent rationality and cohesion mostly supplied by self-delusion, is also common in many alternative AI writings. Alternativists Francisco Varela, Evan Thompson, and Eleanor Rosch relate the disunified self of enactive cognitive science to that understood by Buddhism, stating that both meditation and cognitive science uncover the nonunity of consciousness. Brooks discusses in detail and on scientific grounds why our introspected view of consciousness as unified is fundamentally erroneous ([Brooks, 1995]). In general, the belief that agents can or should consist of separate behaviors with minimal interconnection easily leads to the conclusion that unity, rationality, and coherency are an illusion, or, at best, an emergent property of a fundamentally schizophrenic system.

At heart, antipsychiatrists do not believe that schizophrenics are fundamentally different from other people. As a consequence, they consider 'schizophrenic' as a label to be inaccurately applied to a single person. Rather, antipsychiatrists understand schizophrenia as a social or interpersonal problem; they may claim, for example, that schizophrenics are responding sanely to an insane environment. Fundamentally, they see schizophrenia as an interaction between a person and his or her surroundings. While more recent studies suggest that schizophrenia is not purely or perhaps even largely environmental, the notion that mental illness can be profitably understood by situating a patient in the context of their environment has remained current [Minuchin *et al.*, 1978].

This belief in schizophrenic consciousness as situated in an environment parallels alternative AI's insistence that intelligence can only be understood in terms of environmental interaction. Both antipsychiatrists and alternative AI researchers believe that behavior does not exist in a vacuum. According to this viewpoint, behavior can only be fairly evaluated by understanding it as an interaction between an individual and his or her environment.

4.Schizophrenia as a consequence of a particular kind of decomposition of subjectivity

"With the modern 'psychological' analysis of the work-process (in Taylorism) this rational mechanisation extends right into the worker's 'soul': even his psychological attributes are separated from his total personality and placed in opposition to it so as to facilitate their integration into specialized rational systems and their reduction to statistically viable concepts"([Lukács, 1971], 88)

For cultural theorists, 'schizophrenia' is considered to be both a general way of thinking of people in the 20th century, and a particular and not necessarily positive way of being that is a result of the largely technological and industrialized world in which we live. Schizophrenia is here understood to be a result of living under a system where people are engaged only in terms of one part of their personality; over time, they lose their cohesion as different parts of the personality become autonomous and are no longer coordinated with one another. The paradigmatic example of this kind of schizophrenia is the worker on the assembly line, who may undergo exquisite psychic torture as he or she performs repetitive, mindless motions all day [Doray, 1988].

Schizophrenia in this sense is the limit point of rationalization as it is applied to human consciousness. It is understood as a kind of disintegration that comes about as all qualitative aspects of humanity are eliminated, to be replaced by quantitative, autonomous, and individually rationalized units. With this analysis of schizophrenia as a result of decomposition, we have come full circle: this style of schizophrenia corresponds directly to the technical difficulties alternative AI practitioners face in getting the parts of their autonomous agents to act coherently. Alternative AI practitioners, too, split the ‘souls’ of their agents into autonomous, quantitative units; their agents suffer from the same kind of ‘schizophrenia’ cultural theorists have identified in modern humans. The broader implications of this cultural theory understanding of schizophrenia for alternative AI practice will be studied in greater depth in Chapter 3.

Summary of schizophrenia as metaphor

Each of the metaphors of schizophrenia forms a strand which connects formerly disparate intellectual practices. The strands are summarized in the table below. The advantage of using the term ‘schizophrenia’ is that by studying these strands together, each area has the chance to shed light on the other. At the same time, it is important to note that the usage of schizophrenia in this thesis is not intended to be final. Schizophrenia is not only a metaphor, but also a serious syndrome that affects many people’s daily lives. Its usage here is not meant to make light of their suffering or to suggest treatment options.

From domain	Schizophrenia as...	Corresponds to...
Psychiatry	incoherence	incoherence of behaviors
Philosophy of science	meta-level incoherence	problems in understanding agents
Anti-psychiatry	theory of consciousness	concept of agent
Cultural theory	related to psychological decomposition	behavioral decomposition

Does schizophrenia matter?

Many postmodern theorists have achieved a comfortable notoriety by antagonizing more traditional theorists with their celebration of the virtues of schizophrenia. Simply put, schizophrenia represents for them a liberation from the constraints of behaving as a rational, repressed, neurotic individual. Similarly, many believers in alternative AI celebrate the schizophrenia inherent in their agents. By getting away from a central, hierarchical organization, these scientists feel that they are getting away from many of the flaws of classical AI, and the resultant schizophrenia in their agents becomes a proud marker of their rejection of classical ideas of agenthood.

On the surface, you may find this attitude, if not correct, at least reasonable. Abrupt switching between homogeneous behaviors does not seem such a terrible flaw in the overall scheme of things. Here I will argue that, in fact, schizophrenia can be a fundamental problem, depending on the use to which complex autonomous agents will be put.

The problems schizophrenia raises depend on the use to which you would like to put your autonomous agent. Clearly, for some uses, schizophrenia does not matter at all. If a vacuum-cleaning robot jumps from its vacuuming to its wandering-about-the-house behaviors, this probably does not degrade its vacuuming duties.

Suppose, though, that you want to use an agent as a believable agent, i.e. as a character to which a human is supposed to be able to relate. Believable agents are supposed to allow for a suspension of disbelief, and when they remain too rigidly in one behavior, or switch abruptly between behaviors, they seem unnatural. In addition, agents with a small set of rather shallow behaviors are not so engaging; the user quickly learns to identify the major behaviors the agents can engage in, and then interaction reduces to getting the agent to do one of its 'tricks.' Making agents less schizophrenic means, for believable agents, that the set of behaviors with which the agent is programmed are not so transparently obvious to the user; that the 'parts' from which the agent is built, its behavioral units, blend into a whole personality which invites exploration and discovery without immediately exhausting it.

You may also want to use your agent as a scientific model of a living creature. When we are attempting to build a model that behaves in a similar way to living agents, schizophrenia is something of a problem. After extensive, if not entirely scientific, observation of living agents in the world [Sengers,], I have found it impossible to exhaustively identify the set of high-level behaviors in which the agent engages, and I only very rarely notice abrupt switching between clearly-defined high-level behaviors.

Why I chose not to do a scientific study will become clearer in Chapter 6.

What I *have* noticed is that the very search for high-level behaviors tends to consist of watching a conglomeration of somewhat undifferentiated activity and attempting to come up with plausible explanations about what the agent is doing. What this implies is that the whole notion of 'high-level behavior' is a convenient explanatory mode for identifying gross animal behavior, but that it does not have a necessary detailed correspondence to what the agent is 'actually' doing. The agent may be engaging in a lot of low-level behavior that does not correspond to any high-level behavior, or it may be engaged in some ineffable behavior to which we can simply attach various explanations. When we build scientific models that allow for easy identification of the gross behaviors in which the animal engages, those models are inaccurate in that they display features which living agents do not display, features which are purely a result of the way we built our model. Schizophrenia, not being an attribute of animals in the way we have defined it here, is therefore a problem for scientific agents as well as believable ones.

You may not care about scientific correctness, but simply want to use your agent as a tool. In this case, coherence in the agent is not a value to be achieved for its own sake; it does not bother me, for example, that my text editor switches abruptly from its "writing" to its "printing" modes. However, there are many times when it is not enough for the agent's actions to achieve the user's goal; the user must also be able to understand *why* the agent does what it does. If, for example, a person is teleoperating a semi-autonomous robot, it may be very important that the person can quickly and easily understand what the agent is doing by watching it.[7] If the agent is changing abruptly from behavior to behavior, or switching behaviors so rapidly that the user cannot figure out what the robot is doing, teleoperating it will become much more difficult. Schizophrenia matters for agents-as-tools, because these tools are complex and are often

[7] I am indebted to Red Whittaker for this example.

used by people who need to be able to understand what they are doing.

Finally, you may simply be an AI dreamer who wants to be able to build creatures that are as engaging and interesting as biological beings. If this is the case, you should find schizophrenia very upsetting. Schizophrenic agents do stupid things; they look bad; they reveal the very fact of their mechanicity at every turn. At least a minimal level of coherence is an essential part of what we (albeit perhaps incorrectly) attribute to intentional agents. We will look at this phenomenon in more detail in Chapter 6.

If you are a humanist who does not care about AI, you can sit back and smile politely. Don't worry, your time will come in Chapter 8.

The root causes of schizophrenia

While this formulation of schizophrenia is new, the problem of behavioral coherence has long been recognized in alternative AI. At its most basic, the problem of integrating multiple behaviors per se is foundational. More specifically, building agents that are coherent - that appear to behave consistently across goals and behaviors, not as a bundle of parts — is an explicit goal for many researchers. Brooks ([Brooks, 1994]), Blumberg, and Loyall, for example, all mention this kind of apparent behavioral coherence as a goal of their work.

"Even though many behaviors may be active at once, or are being actively switched on or off, the robot should still appear to an observer to have coherence of action and goals. It should not be rapidly switching between inconsistent behaviors, nor should two behaviors be active simultaneously, if they interfere with each other to the point that neither operates successfully" ([Brooks, 1991a], 22).

These researchers have put a lot of work into trying to understand how to design and build coherent agents. Loyall has, for example, developed agent design strategies and architectural support for mixing multiple activities in pursuit of a goal, so that an agent does not, for example, freeze in place while it is trying to decide what to say. Blumberg has also put substantial effort into addressing coherence. His system addresses the problems of rapid switching and multiple conflicting behaviors, and he has some novel techniques for combining simultaneous behaviors. However, the problem of abrupt behavior switching remains, and is, if anything, more clear in his systems than in the others. Fundamentally, these solutions, while chipping away at particular symptoms of schizophrenia, do not address the fundamental problem that behaviors are designed separately, and that the boundaries between them become clear in the activity of the agent.

Given this recurrent interest, the inevitable conclusion must be that the problem of schizophrenia has not remained unsolved due to a lack of interest, effort, or talent. Why is this problem so hard to solve? To put it at its simplest, behavior-based AI runs into the same problems classical AI has — if you divide your agent into parts, it is natural to have problems integrating those parts back together again. But the problem is deeper than finding some ad hoc solution to hook up the disparate parts of any particular architecture. The claim I will make here is that *schizophrenia has not been solvable because it is an inevitable result of the current agent design process*. In order to understand this, we will need to take a closer look at how we construct agents.

Agents as atomized

Unlike biological agents, artificial agents begin life as a concept in their designer's head. At this stage, an agent is an idea — a living agent to be copied, a dreamed-up character, a potential solution to a problem — which is analyzed in order to yield the constituent parts that become the eventual agent's behaviors.

"There are many possible approaches to building an autonomous intelligent system. As with most engineering problems they all start by decomposing the problem into pieces, solving the subproblems for each piece, and then composing the solutions." ([Brooks, 1986b],6)

For example, imagine that you want to build an artificial dog. To do this in the behavior-based manner, you will first need to decide what the basic behaviors of the dog are. You generally do this by looking at the dynamics of the agents' activity, and trying to recognize what the parts of that activity are. Thinking about or watching a dog, you might come up with some typical behaviors: eating, playing fetch, sleeping, etc. After you've come up with the behaviors, you connect them using your architecture's default behavioral organization mechanism. In the end, you might end up with something like Figure 2.7.

Behavior-based agent design works by breaking the dynamics of the imagined or observed interactions into parts by parsing the dynamics of the agent's behavior for meaningful subunits. That is, the behavioral units chosen for the agent are a result of an interpretation of the imagined or observed agent's interactions with a user or environment. This interpretation is fundamentally symbolic; as in parsing, the agent's behavioral dynamics are divided into meaningful, somewhat independent units.

If you have a humanist background, you may recognize atomization as a form of reification, applied to objects of scientific study. This view of atomization will be explored in more detail in Chapter 3. Atomization is also similar to reductionism, the belief that objects are made up of the simple combination of simpler objects. Atomization is, however, not necessarily a statement about the way the world is organized; it can simply be a way of approaching phenomena in order to make them easier to understand.

This process of splitting-up I term *atomization*. Strictly speaking, atomization refers to the process of splitting something that is continuous and not strictly definable into reasonably well-defined, somewhat independent parts. The term *atomization* comes from neurology [Goldstein, 1995], where the *atomistic method* refers to a method of trying to divide the brain into small, localized pieces, each of which corresponds to exactly one behavior. The use of atomization in computer science, under such watchwords as modularity, decomposition, and divide-and-conquer techniques, has been more successful. These techniques form the core of programming methodology and are essential tools for making large systems that people can design and understand.

In fact, methodologies akin to atomization are not limited to computer science. The advantages of using atomization to understand complex systems are understood in many sciences. It has similarities, for example, with the digitization of analog signals, with dissection of organisms in anatomy, with the identification of species in population biology, with the classification of mental illness in psychiatry, and, in general, goes hand-in-hand with formalization and analysis. In all these cases, atomization is a way of getting a handle on a complex phenomenon, a way of taking something incoherent, undefined, and messy and getting some kind of fix on it.

It should be clear at this point that a fundamental tenet of behavior-based AI is behavioral atomization. Manifestos on behavior-based AI regularly cite behavioral decomposition into independent units with limited interaction as one of the defining characteristics of the movement:

> An agent is viewed as a collection of modules which each have their own specific competence. These modules operate autonomously and are solely responsible for the sens-

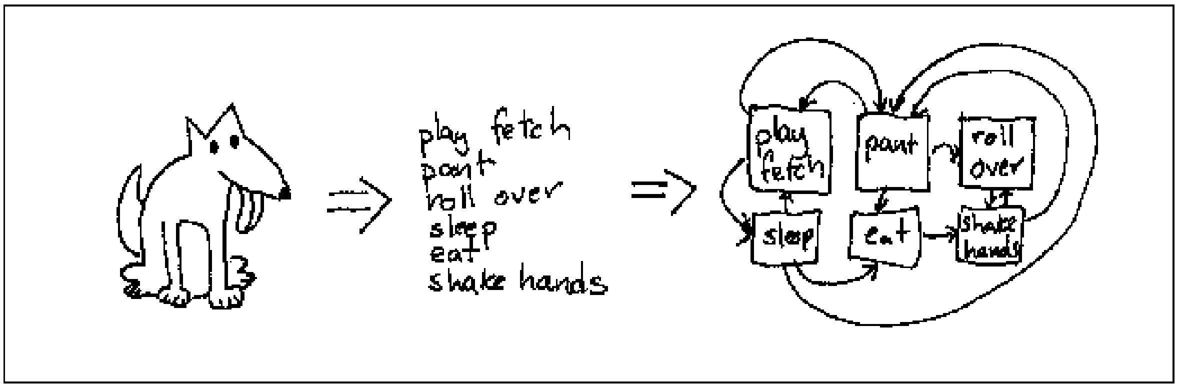


FIGURE 2.7: Design steps for an artificial dog

> ing, modelling, comptuation or reasoning, and motor control which is necessary to achieve their specific competence.... Communication among modules is reduced to a minimum and happens on an information-low level. ([Maes, 1990b],1)

Alternative AI researchers are generally allergic to the concept of central control; one generalization to which this not infrequently leads is that behaviors should be designed and built separately and refer to each other as little as possible.

Special properties of atomization

In alternative AI, atomization is a process of breaking down a behavior into meaningful units closely akin to the process of parsing natural language. There is, however, a major difference between parsing natural language and understanding an agent's behavior; while in listening to native language speakers we can be reasonably certain that the stream being parsed truly contains symbols, it is unclear in what sense we can truly say an agent's physical presentation is a more or less linear stream of clear-cut behaviors. When observing living creatures, for example, one can certainly deduce a set of high-level behaviors [Benyus, 1992], but one would be hard-pressed to even identify every movement of an animal as being part of one of these behaviors, let alone understand all behavior purely as a succession of these well-defined, a priori behaviors. Given the mess that is the nervous system, it's hard to even imagine how such a neat, tidy behavioral presentation could ever happen.

> "Real biological systems are not rational agents that take inputs, compute logically, and produce outputs. They are a mess of many mechanisms working in various ways, out of which emerges the behavior that we observe and rationalize." ([Brooks, 1995], 52)

Atomic behaviors, then, are not pre-given — they do not exist in the world per se. Rather, these atoms are an interpretation of agent activity, distilled into units which carry meaning for the observer. Atomization is a kind of explanation, a process of understanding that comes about as we try to bring order to our experience of the world. In this sense, atomic behaviors are not what the animal does, but our best explanation to ourselves of what the animal is doing. Atomization is a form of approximation, taking noisy, messy, real-world activity and distilling it into a more formal and clean representation.

This does not mean that atomization is arbitrary or useless. Atomization brings with it properties that are valuable; the representations it generates are essential to helping us understand and engineer behavior. Atomization is essential to science because it brings order to inchoate experience, giving you pieces out of which a formal system can be built. Atomization is, in fact, also endemic to the humanities (through theoretical categories such as race, gender, and class distinctions, literary periods, and genre), although the use of atoms there is somewhat different because of a greater ingrained skepticism for the absolutism of categories and because of a different conceptualization of how atoms relate to one another and to the phenomenon they model (we will build on this in Chapters 3 and 6). In both science and the humanities, atoms give you the basic units out of which meaningful understanding of the world can be constructed.

Atoms are not, however, simply transparent lenses through which the world is viewed. The atoms out of which we build our agents have their own special properties, which form the basis of our ability to use them to understand and build artificial agents:

- Atoms are **discrete**. Natural behaviors generally blend together, making it hard to define a clear moment when an animal changes, for example, between being asleep and being awake. Atomic behaviors partition these analog behavioral changes into clear states. There are no in-between states, no processes of transformation between one behavior and another, no moments when the action of the agent cannot be attributed to one of the labelled behaviors at all.

- Atoms are **meaningful units** of action. Atomic behaviors correspond to activities that make sense to the observer / designer. In this sense, behaviors are *symbolic*. They are conceptual chunks of the agent's activity.

- Atoms are **cleaner than real-world behavior**. Real-world behavior is messy, not always clearly definable or understandable. Atomic behaviors clean up this mess, allowing us to build systems that are understandable, programmable, controllable.

Atomization, then, is fundamentally *the reduction of an observed, analog stream of activity to discrete, meaningful, symbolic parts*.

Schizophrenia as a consequence of atomization

Because atoms have their own special properties, an agent that we build from atomized units is in important ways not equal to the thing it reproduces. An atomized agent consists of a symbolic representation of the original creature's actions. There is nothing wrong with this situation per se; it is a simple statement of the fact of the agent's construction. In order to understand the creature's actions, we create a symbolic representation of those actions; it is these symbols that form the basis for the engineered reproduction of the agent.

What happens when we build our agents from these symbolic units? The tendency is for one of two things to happen: either the behavior is completely incomprehensible to the user, or, to the extent that the behavior is comprehensible, the user can recognize the behaviors with which we programmed the agent. Since, generally speaking, we intend for users to be able to recognize these units, this recognition is exactly what we wanted. And since the behaviors we chose are precisely what we found meaningful in the original creature's behaviors, it is not surprising when the interactor can recognize them in the agent as well. We are happy if we have succeeded, and the user knows it is hunting, playing fetch, etc.[8]

There is, however, a problem, in that all the actions of the agent are a result of these symbolic, hopefully recognizable behaviors. The agent's behavior, if understandable at all, becomes so clear-cut that it is amenable to a kind of parsing on the part of the observer that is unreasonable for living creatures. This is what causes them to seem unnatural. While with living creatures there is always some amount of 'noise' (the extent to which the atomizing approximation is only an approximation), artificial creatures are all high-level, potentially understandable, symbolic behavior. People quickly notice the categories into which the agent's activity is divided; they can see that, unlike biological agents, this creature is pure representation. This, then, is the source of schizophrenia in agents: the modularity of agent design into symbolically meaningful units means that the individual behaviors of an agent are too clear-cut. Agents jump from behavior to behavior in a jarring and often meaningless sequence.

At this point you may come to the conclusion that the way to solve this problem is not to have any explicit behaviors. This is in fact the solution used in architectures like Pengi and ANA. Agents built in these architectures do not exhibit any pre-planned behaviors per se, but rather mix together actions from different behaviors according to whatever seems logical at the moment. Interactors certainly will not recognize behaviors if there are no behaviors to recognize.

But this does not fundamentally solve the problem of schizophrenia. Agents do not engage in clear-cut behaviors, but mix together actions from different behaviors. Still, each action the agent chooses is from a particular, designed high-level behavior. While Pengi and ANA allow the agent to interleave actions — choosing actions alternately from different behaviors — they do not allow agents to engage in action that is not directly related to one of the designer's chosen high-level behaviors. This also means there is no mediation, averaging, or transformational processes between behaviors. The agent can only take action that is logical within the parameters of one of its behaviors.

In addition, in these architectures each action the agent takes is chosen because of its logic for a particular high-level behavior. Since actions are chosen for their logical role in separately designed behaviors, it is likely that they will make less sense in the agent's 'emergent' behaviors. In particular, strange behavior will result any time the logical structure of the two high-level behaviors is different. Since the philosophy of behavior-based AI is to design behaviors as separately as possible, this state of affairs is bound to happen regularly.[9]

[8] In addition, we are also sometimes happy if the user comes to think the agent is doing other intelligent things that, strictly speaking, we haven't programmed it to do. The phenomenology of projected behavioral identification would be an interesting subject for another thesis, to which I think Chapter 6 provides some initial clues.

[9] It is also not clear that the approach of behavior-less behavior construction scales well to multiple, complex behaviors. It may be that truly complex behavior of the kind required to run an articulated graphical or robotic agent with a wide range of activity requires structure like behaviors for the programmer to be able to keep track of what is going on.

Atomized behavior, then, if comprehensible at all, will display one or both of the following attributes, depending on how it is synthesized:

1. It may be synthesized in terms of symbolic behaviors. In this case, it will tend to jump abruptly from behavior to behavior. These behavioral switches will be apparent and jarring to the user, since users, like designers, will understand the agent's activity in terms of symbolic activities.

2. It may be synthesized at the level of actions, with actions chosen according to the logic of the separate symbolic behavior of which the designer sees that action as a part. In this case the actions will often be mixed in a way that violates the individual behaviors' logic, resulting in an incoherent, nonsensical jumble of action.

In any of these cases, the resulting agent will display the symptomatology of schizophrenia as I defined it earlier. My conclusion is that *schizophrenia is a direct and inevitable result of atomization*. It is a fundamental property of our agent design strategy.

The catch-22

If schizophrenia is caused by atomization, then it would seem that the most obvious way to get rid of it would be to get rid of atomization. This is, in fact, the agent design strategy proposed by Loyall: all the parts of the agent should be designed together. But this solution is impractical for large, complex agents, as we discovered when we built the Woggles. Atomization is an essential strategy for simplifying phenomena enough that we can understand them. Getting rid of atomization means understandable, modularized code is thrown out the window. Making behaviors arbitrarily complex and interrelated also makes them arbitrarily difficult to debug and comprehend. For the sake of being able to program, we need a certain amount of atomization.

At this point we are backed into a corner. The final conclusion of the arguments made here is that atomization causes schizophrenia, but we need atomization to write code. This is a vicious circle.

If this argument holds, then schizophrenia will not be solved by a clever new algorithm. It then represents the absolute limit point of current ways of understanding agents. As far as I can tell, *schizophrenia cannot be addressed within current AI frameworks*. It is a dead end.

The goal of this thesis is to change this. I believe AI can and should be done differently. This will require us to rethink the foundations of AI. Such rethinking has traditionally been done through importations from the sciences and analytic philosophy. While many of these importations have been ingenious, insightful, and stimulating, I suspect they are not enough, since they generally share the same atomistic principles. To solve schizophrenia, I believe we will need a radically new perspective. This thesis explores the possibility of getting that perspective by understanding AI as culture.

In the next chapter, we will use this humanistic approach to come to a deeper understanding of schizophrenia and its relationship with intentionality. Cultural theory and antipsychiatry make some suggestions about how schizophrenia can be handled. This different way of handling schizophrenia will become the basis for the technical results in the second half of the thesis.

Chapter 3

Schizophrenia, Industrialization, and Institutionalization

This chapter, and indeed this entire thesis, has its origin in a week in 1991, which I had the pleasure of spending hospitalized for depression at Western Psychiatric Institute and Clinic in Pittsburgh. This week was a turning point in my life, not because of any particular help I received in my hour of need, but because, for the first and hopefully last time in my life, I discovered what it was like to spend 24 hours a day under the surveillance of a scientific system that tries to regulate every aspect of individual, human, subjective experience. The amazing paradox that became clear over my days in this fishbowl environment is that the more subjective experience is placed under surveillance, categorization, and attempted control, the less it is actually observed, understood, and influenced. Using a label like 'atypical personality disorder' and writing a prescription for Anafranil did little to address the existential crisis that had brought someone to this unbearably painful point in life. Instead, it tended to build barriers, to separate patient from doctor, to make the doctor feel competent to judge the humans in his or her care as instances of a category, and to keep the doctor from being drawn into the muddled details of treating him or her as a complex, messy, fellow being.

This experience made clear for me the limitations of objective approaches to understanding subjective experience. Certainly, objectivist[1] knowledge traditions such as psychopharmacology have their place; no one can deny, for example, the power of lithium to give manic-depressives stability in their lives. At the same time, these objectivist traditions, particularly when seen as the only and ideal goal of all human intellectual endeavor, leave something important out: individual human experience, with all its rich and ambiguous implications, with its meanings not objectively available but to be sorted out moment-by-moment by specific

[1] By 'objectivist' I mean "having the goal of objectivity." Whether these forms of knowledge production can ever actually achieve true objectivity (whatever that is) is far beyond the scope of this thesis, though my guess would be that, since they are so bound up in interpersonal relations and cultural traditions of what is and is not normal, they probably can't.

people in concrete situations, experience which is most adequately jointly approached not as doctor-patient, not as subject-object, but as equal and unique but related individuals leading complex and very real lives.

In this chapter, we will look at the limitations of objective knowledge and possibilities for subjective understandings of human life from various perspectives. The goal is to flesh out the understanding of schizophrenia and its relationship to atomization that we started exploring in Chapter 2. We will do this by looking at two particular case studies that relate to AI's ways of understanding agents: industrialized understanding of workers on the assembly line and psychiatric understanding of mental patients. In both of these cases, we will see similar processes at work: the attempted categorization and control of sometimes frighteningly ineffable human behavior through the application of objectivist forms of knowledge production. In each of these cases we will also find limitations in the extent to which these techniques can actually be used to control and understand human behavior; subjectivist critiques of each tradition will provide alternative ways of understanding that may be more helpful.

Dear technical reader, this chapter may be quite a challenge to you (though it may also be an unaccustomed treat). Although the conclusions of this chapter will form the basis for the technical results in the rest of the thesis, little that I say in the pages of this chapter will bear directly to problems in autonomous agents. The approach in this chapter will be exclusively humanistic. The form of argumentation will be largely metaphorical; I will try to draw out metaphorical connections between various cultural practices that relate to schizophrenia in autonomous agents. There is a logical argument to be found here, but in the rhetorical forms of cultural theory, the 'point' is not only the argument but also the details of the concrete situations that are looked at. "What matters above all is not to reduce everything to a logical skeleton, but to enrich it, to let one link lead to the next, to follow real trails, social implications" ([Guattari, 1984], 259). Fundamentally, this chapter is selling, not an argument, but a particular and rich way of seeing which can apply to various parts of life, including but not limited to AI. Good luck and enjoy!

Case Study 1: AI as Industrialization

When I took my first course in AI, I was gripped by an intense fascination: why on earth would anyone think AI was even possible? With all the amazing, strange, wonderful, horrible, bizarre things that humans do and are, what would ever possess anyone to think that this miraculous existence could be reproduced by a machine? How can indefinable, ungraspable consciousness be thought to be 'implemented' in machinery, apparently as a set of search algorithms? What kind of strange and twisted view of humanity is embodied in chess-playing machines as philosophy of life?

After being immersed in the field for several years, I began to see AI as a natural view on life; it becomes hard to remember this initial sense of wonder. In fact, AI researchers tend to feel that any mention of it is a sign that the bearer of the wonder is either a fuzzy-headed believer in the supernatural or suffering from a little heat stroke that a good nap might cure. But even if we are willing to grant AI an intellectual certainty, a question still lurks: why is it that, at this moment in history, AI as an intellectual endeavor seems to so many not only possible but self-evident? What is it about our current way of thinking that makes the very idea of AI a natural extension of our intellectual traditions?

In and of itself, the idea of reproducing life is nothing new; medieval Jews already had the tradition of the Golem, an effigy magically brought to life; the 19th century brought the organic horror of Frankenstein. But the techniques by which AI approaches the problem of creating artificial life are different; the creation of life is no longer a question of magic, alchemy, or biology, but one of information. Artificial beings are not made of clay or rotting body parts, but of algorithms glued over bits of silicon and robotic machinery. AI and cognitive science approach life not as a mysterious spiritual or biological process to be engaged in or mimicked but as a machine, like any other, to be designed and controlled. AI, in this sense, is the next step of industrialization: having replaced worker's bodies with robotic machinery, we are now developing replacements for the worker's minds. In this section, we will look at AI as the industrial mechanization of subjectivity. We will dig deeply into industrialization to understand AI's inheritance from it: techniques, philosophies, but also problems, among them schizophrenia.

"In a sense, the mechanical intelligence provided by computers is the quintessential phenomenon of capitalism. To replace human judgement with mechanical judgement — to record and codify the logic by which rational, profit-maximizing decisions are made — manifests the process that distinguishes capitalism: the rationalization and mechanization of productive processes in the pursuit of profit.... The modern world has reached the point where industrialisation is being directed squarely at the human intellect."([Kennedy, 1989], 6

Industrialization as Mechanization

The history of the Industrial Revolution is, among other things, a story of the gradual replacement of workers by machines. This fable proceeds as follows: in the beginning there were craftspeople, who owned their own tools, who manufactured articles in their own, idiosyncratic ways, whose work was largely integrated with their way of living. As the Industrial Revolution begins, these workers begin to be collected into factories, where they work together using the owner's tools. This owner, in an attempt to make work more efficient, begins to streamline the production process. Instead of having each worker build a piece from beginning to end, the production line is developed, where each worker works on some small part of the final piece. Work is broken up into stages, each of which is accomplished by a single worker; each stage is standardized so that articles can move from stage to stage without breaking work rhythm. Once work is divided up into standard stages, some of the steps can be done by a machine. Instead of building an article from beginning to end, workers now tend machines which are each doing small steps of the article's production.

At each stage, work becomes more rationalized, predictable, and efficient. Workers on the assembly line can generate more articles, and the articles lack the idiosyncratic variation of normal craftwork. Instead of doing whatever he or she wants in a haphazard order, a worker has a fixed set of steps he or she engages in. The intelligence of the worker, which s/he previously needed in order to monitor what s/he was doing and make active decisions about how work should proceed, is now embodied in the structure of the assembly line. Workers no longer need to think; the factory machinery does the thinking for them. Even before computers, industrialization takes the first baby steps of AI.

These traces of industrialization can be seen in the way we build agents today. The AI researcher building an agent follows the same basic line as the factory manager designing new production processes. Just as the factory manager attempts to design and reproduce a pre-existing work process, the AI researcher would like to copy a natural process — an agent or idea of an agent. Just as the factory manager breaks this process into logical steps, figuring out which steps should happen and in which order, AI researchers analyze the agent's behavior, to categorize its activity into typical behaviors and to enumerate the conditions under which those behaviors are appropriate. And just as the factory manager embodies each step in machinery which can run with a minimum of human supervision, AI designers implement a mechanical version of each behavior, hooking them together so that they largely reproduce the imagined or real behavioral dynamics of the original creature. The early industrialist and the AI researcher are engaged in the same project: we analyze, rationalize, and reproduce natural behavior.

At first blush, a difference between AI research and industrialization may seem to be that AI seeks to reproduce intelligence, whereas the industrialist is not so much interested in reproducing work processes as in maximizing profit on their output. This means that post-industrial work is radically different from pre-industrial work, in both the qualities of the articles produced, and in the human experience of engaging in that work. The very act of embodying work in the production line changes the nature of work. Work becomes more rationalized and less personal; workers are more dependable and more bored; the articles produced become more standardized and less individual.

But in Chapter 2, we saw that, just like early industrialists, AI researchers do not create absolutely faithful reproductions of the living beings they seek to emulate. We looked at the special properties that artificial creatures have when they are built using atomized processes. Atomization, we learned, introduces its own qualities that can be recognized in the creatures generated with it, among them schizophrenia.

Similarly, several special qualities of post-industrial work and life have been identified by cultural theorists and industrial historians:

- *Reification* — Things that were once thought of as ineffable or abstract become thought of as concrete. 'Labor,' for example, which was once not strictly separated from the rest of life, becomes something that is measured and sold per piece or per hour. Once things are reified, they can be sold, becoming *commodified*, to be exchanged for particular sums of money.

- *Specialization* — Workers no longer engage in the entire work process; rather, they each perform some small function within the process as a whole. Without an overview of the process, workers no longer need or are able to adapt to one another; each part takes place without reference to the others. Without feedback between the pieces, each piece is built in isolation, the whole then being merely the sum of each individual, separately designed atomic part.

- *Atomization* — The production process, which was once a wholistic attribute of individual workers, is broken into rationalized parts,

Marxism: a Demystification

Since the beginnings of the Cold War, Marxism has had a bad name in the United States. Its use by Communist totalitarian systems has not done much for its image in the public eye. Since the fall of the Soviet Union, much of the educated public has been led to believe that Marxism, like Communism outside of China and Cuba, is dead.

But Marxism is not just a (seemingly failed) political doctrine predicting the end of capitalism, but also a thriving intellectual tradition. This tradition includes some of the greatest thinkers of the 19th and 20th centuries, most notably but certainly not limited to Marx himself, an intellectual and scholar whose influence has been felt around the globe and in many disciplines for a century and a half. In this tradition, Marxism can be understood simply as a theory of specifically industrial society; this face of Marx forms the basis not only of scary left-wing political theory, but also of much of modern economics.

It is impossible to do any serious analysis of industrial culture without Marxism, and this thesis is no exception. This means a continuing dialogue with the Marxist tradition, one that takes into account not only Marx's original writings, but also contemporary reinterpretations of them. Most notably, it seems that Marx's prediction that capitalism would lead to its own downfall is probably incorrect; and nearly all Marxist-influenced thinkers consider the reduction of all cultural activity to class warfare as long since passé. Nevertheless, that still leaves plenty of grist for the intellectual mill. Here, I will focus particularly on Marxist analysis of the changing experience of being human as more and more kinds of labor become mechanized.

each of which is embodied in pieces of machinery or in production rules that regulate how they interact. Workers, who were once thought of as individual humans deeply embedded in the context of their daily life, now become interchangeable parts of the production process, whose time is to be sold to the highest bidder. They move from factory to factory, no longer connected to their home place or even to a particular manufacturer. Workers see themselves as free and atomic individuals, bound by no human ties.

- *Standardization* — The idiosyncrasies of craftwork means that one can never be sure what the produced goods will be like. The factory owner, on the other hand, who consolidates craftwork and has promised broader distribution networks goods of a particular kind and quality, wants to have some guarantees that the factory will produce similar goods no matter which workers are present on a particular day. The idiosyncrasies of personal work are no longer valuable; instead, the owner introduces steps of production control to make sure that the output is always similar. The qualitative, human, individual dimension of work is eliminated, replaced by efficient, controlled, and standardized work processes.

- *Formalization* — The individual, material properties of workers and the material they operate on is ignored, except insofar as it impinges on the production process. As the production process becomes more and more efficient, extrinsic considerations — whether social, spiritual, or physical — are left out. The production manager thinks of the production process in terms of abstract steps, without reference to the particular identity of the worker or chunk of material involved; the factory is set up to enforce this abstract, impersonal view, which then seems to be an accurate reflection of the real. Individual differences become 'noise,' unvalued and only reflected upon in order to control their effects.

- *Mechanization* — In order to maintain standard production, workers are given less and less leeway in decisions about their jobs. Rather than relying on the worker's judgment, the factory manager uses standardized production rules to ascertain that the product is always made the same way. As the steps of the production process are more and more formalized, the worker's intelligence becomes less and less pertinent. Once the worker's intelligence is no longer needed, the worker can be replaced by a machine.

These trends in industrial culture are rooted in factory work, but they did not stay confined to the factory for long. Workers, who spend a large portion of their waking hours interacting with machinery on the production line, take home the values that that system has ground into their bodies. Production line designers, factory owners, and managers, spending their days designing machinery and optimal control of the human-machine interface, do not always forget their machinic view on life on their days off.

More insidiously, the drive of the assembly line, powered by the money its efficiencies can bring, spreads into other intellectual fields: factory owners hire inventors, scientists, and engineers to design machinery and the production processes they support; they hire social scientists and management experts to design worker compliance (Total Quality Management is born). Each of these fields, applying itself within the context of the assembly line, starts to find more and more ways to generate interesting results within the factory work framework, slowly and mostly unconsciously taking over the assumptions of formalization, atomization, and so on that that framework presupposes. Factory owners lobby for laws that support and reflect their point of view on factory work. Both private and public institutions are set up to explicitly market these practices; the military, for example, played a large hand in encouraging development of standardized manufacturing [DeLanda, 1991]. Other businesses, which are not strictly factory-oriented, envy the efficiency and rationality of factory work, and begin to apply some of their ideas to improve their own processes. Soon the countryside is dotted with identical, standardized, efficient fast food restaurants with its teenaged automatons taking on the role of factory machinery; no one living in these cultures can escape the force of industrialization, even on their lunch break.

All this talk of workers in the factory and the assembly line may come across as antiquated today. How many of us are still factory workers on the assembly line? When 'us' means the readers of this thesis, the answer must be very few. After all, we late 20th century Westerners are no longer in the industrial era; we are brave new citizens of the Information Age![2]

But the forces of industrialization, far from having disappeared, have become so ingrained in our daily lives that they are taken for granted. If you live in the West, and especially if you are American, industrialization is the air you breathe and the prepackaged food you eat. Your life becomes more and more mechanized as your bank teller is replaced by an Automated Teller Machine, your receptionist is replaced by a voice mail program, the telephone solicitor who has interrupted your dinner every night for the last 6 years is replaced by an auto-dialer with a cheery robotic recording. The last bastions of your craftwork slowly give way as universities become digital diploma mills, offering impersonal and standardized distance learning to students who are no longer limited by the bonds of location or social interaction [Noble, 1998]. When in Paris, you eat your standardized lunch at McDonald's, knowing that, while it may not be very good, it won't expose you to the idiosyncrasies of local cuisine — any calf brains will be ground beyond recognition into your Big Mac. You reify yourself as you sell 4 hours a day to each of 3 part-time jobs, trying to still maintain a full sanctified hour of Quality Time with your youngster — go ahead, sell yourself until you can afford to buy yourself back! You have specialized yourself, become the world expert on polynomial kernel support vector machine with fractional degree, unable to discuss your work with more than 3 or 4 colleagues because it is so hopelessly obscure (but nevertheless breathtakingly important). You atomize yourself, cut off from your extended family, perhaps even from your spouse and children, moving every 7 years in an evanescent search for the better life. How many times a day do you formalize yourself, jacking into cyberspace to become blissfully unaware of the constraints of your undeniably material, geographically located, and mortal flesh — at least until your RSI kicks in?

[2]Note that life looks very different to those in the third world, for whom underpaid and dangerous work on the assembly line is still a very real daily experience.

No, industrial culture is not confined to the 19th-century factory; it continues to live itself through us on a day-to-day basis. Industrial culture is not just an attribute of a now marginal work-life; it colors nearly every aspect of late 20th century Western existence [Strasser, 1982]. It is not just a way of producing goods, but a new and not always positive way of being. We are post-industrial humanity: reified, specialized, atomized, standardized, formalized, mechanized; we are nonstandard flesh, the weak link in a network of machines.

"[T]oday Western man has become mechanized, routinized, made comfortable as an object..." ([Josephson and Josephson, 1962], 10)

Taylorism and Schizophrenia

What is it like to live a post-industrial existence? For humans, industrialization is often an experience of being more and more dominated by systems of machinery, of both the technical and bureaucratic kinds. This is certainly the case for craftworkers, whose work historically consisted of skilled tinkering in the workshops of their houses, but presently generally involves the monitoring of raw materials as they are fed through massive machinery. Rather than applying their intelligence and skill to an ever-renewed activity, taking pride in the result of their handiwork, workers go through repetitive and mindless motions that are stipulated from beginning to end by the production handbook in order to create finished products they will never see. The experience of being a worker was once work; now, it is being an appendage to a machine.

"Taylorist man is a slave to the movements of a machine, and he cannot control it either technically or socially. Above all, he suffers from the divorce between that part of his body which has been instrumentalized and calibrated and the remainder of his living personality." ([Doray, 1988], 82)

In a sense, it is workers themselves who have become mechanized. Georg Lukács has shown that the mechanization of the work process does not stop with production itself; the workers themselves are progressively designed and controlled as machines.

> If we follow the path taken by labour in its development from the handicraft via co-operation and manufacture to machine industry we can see a continuous trend towards greater rationalisation, the progressive elimination of the qualitative, human and individual attributes of the worker. On the one hand, the process of labour is progressively broken down into abstract, rational, specialized operations so that the worker loses contact with the finished product and his work is reduced to the mechanical repetition of a specialised set of actions. On the other hand, the period of time necessary for work to be accomplished (which forms the basis of rational calculation) is converted, as mechanisation and rationalisation are intensified, from a merely empirical average figure to an objectively calculable work-stint that confronts the worker as a fixed and established reality. With the modern 'psychological' analysis of the work-process (in Taylorism) this rational mechanisation extends right into the worker's 'soul': even his psychological attributes are separated from his total personality and placed in opposition to it so as to facilitate their integration into specialised rational systems and their reduction to statistically viable concepts. ([Lukács, 1971], 88)

Taylorism, or scientific management, is the apogee of this view of worker-as-machine. The goal of Frederick Taylor's scientific management is to increase the efficiency of work processes by analyzing and optimizing not only the machinery itself, but also the way in which the worker uses the machines. Through time and motion studies, the worker's motions are examined; all motions are forbidden. Rather than letting workers interact with machinery in any way that they saw fit, Taylorists determine the "one best way," the most efficient possible use of the machinery. Upon Taylorization, workers are generally given detailed instructions of every movement they should use to accomplish their job. Nothing is left to chance; nothing is left to worker ingenuity; nothing ever varies. With Taylorism, the rationalization of the work process, having extended into the worker's mind, is complete.

"It is not simply status-hunger that makes a man hate work that is mindless, endless, stupefying, sweaty, filthy, noisy, exhausting, insecure in its prospects and practically without hope of advancement. The plain truth is that factory work is degrading" ([Swados, 1962],111).

Despite the radical successes of Taylorism in improving the efficiency of industrial work, it also has some unexpected negative effects. Taylor thought that workers would be happy to be able to work more efficiently and make more money. Instead, unions object to Taylorist techniques because they reduced workers to mindless objects, ignoring the expertise of skilled workers in favor of scientific analyses by outside experts. Workers find the absolute banalization of the work process that Taylorism implies unbearable; Taylorist work is both repetitive and mindless, on the one hand wearing out workers' bodies with repetitive stress injuries, on the other boring them senseless [Doray, 1988].

Taylorism demands that, not only the process of production, but humans themselves become rationalized and mechanized. The difficulty in this plan is that people are not machines. While Taylorists are able to categorize and optimize worker movements, they do so while ignoring the worker as human being. The result is that a small part of the worker's existence is identified and reinforced; the remainder is repressed, until ignored aspects demand attention when the worker is injured, becomes distracted, or simply refuses to submit to such a repressive regime (or, in the case of postal workers, shoot their co-workers and bosses).

Ironically, while Taylorism leaves something to be desired for its original goals, it is extremely well-suited as a basis for Artificial Intelligence. While workers cannot handle these repetitive, mindless activities, they are perfect for robots. Numerous scholars have pointed out that Taylorism is the last intellectual stop before AI: as soon as work is reduced to mindless, rote movement, idiosyncratic and moody human workers can be replaced by controllable and indefatigable robots, removing the last unpredictable part of the production process.

The principles of Taylor — quantifying and rationalizing human behavior, reducing intelligent behavior to a set of independent, predictable, and interchangeable parts, removing all traces of human idiosyncrasy, creativity, and craftwork — are now suspect in management, but live on in the engineering tradition in computer science. Michael Mahoney, a historian of science, points out with some surprise that in software engineering, the arguments are not about whether the principles of Taylor are correct, but about how to apply them [Mahoney, 1997]. This observation extends to Artificial Intelligence — in many ways, AI is simply the late 20th-century reincarnation of turn-of-the-century traditions of human engineering and control.

In Taylorism, as in agent design, a coherent and wholistic behavioral dynamic is carved into chunks. Individual pieces of behavior are identified and rationalized. In Chapter 2, we noted that this partial rationalization leads to schizophrenia, and the same effect happens under Taylorism. Again, this schizophrenia is not meant as a psychiatric label (although it certainly seems possible that assembly line work could drive someone insane); by schizophrenia I mean a disintegration of subjective experience as some parts of a person are brought out and others repressed. This schizophrenia is experienced directly as the boredom, degradation, and depersonalization of assembly line work.

Schizophrenia for the Masses: Industrialized Life

"Our society produces schizos the same way it produces Prell shampoo or Ford cars, the only difference being that the schizos are not salable." ([Deleuze and Guattari, 1977], 245)

This form of schizophrenia is not simply a result of Taylorization, although certainly Taylorism displays it more extremely. Marxist and post-Marxist scholars understand this kind of schizophrenia to be a result of simply living in industrial society [Lukács, 1971] [Deleuze and Guattari, 1977]. This is because all of us in post-industrial society — whether assembly line workers, hamburger flippers, or university professors — are constantly coming into contact with machinery and bureaucracy that is set up to ignore most of what we might value in ourselves. We are enmeshed in objective 'laws,' imposed from outside: the rules of the assembly line, the invisible hand of the market, the laws of physics. We live our lives qualitatively, while continuously being asked to make decisions and define ourselves in terms of quantitative and inhuman systems. These mindless systems come set up with a priori categories; our freedom and humanity is manifested only in that we can choose which category we want to be processed through. The industrialized doctrine of individuality is "choose 1 of *n*:" you can choose one of 6 Extra Value Meals, drive your sport utility vehicle to one of 14 suburban malls, click on one of 8 links, buy one of 123 kinds of cereal, punch in one of 9 responses to the voice-mail prompt, vote for one of 3 politicians, identify with one of 4 ethnic groups; but if you want to stay in the system you cannot meaningfully choose 'none of the above,' or, God forbid, half of one and a third of another with a little bit of something extraneous mixed in.

George Ritzer studies the extent to which themes from assembly line work, Taylorism, and bureaucracy have infiltrated our daily lives [Ritzer, 1993]. This 'McDonaldization' of society is based on the growing cultural importance of *formal rationality*, i.e. systems under which people try to find the best ways to achieve pre-given and unquestioned goals, not according to their personal feeling for how it should be done, but by reference to formal systems of rules and regulations. This kind of rationality is interested, like Taylorism, in "the one best way" to do things, and is suspicious of the ability of individual people to judge things for themselves. Instead of leaving decisions up to the people involved, technical, legal, and bureaucratic systems are set up so that the 'best' way to do things is also the natural way. In industrialized society, 'best' is judged along four axes:

- *efficiency* — The production line maximizes the efficiency of craftwork; everything extraneous to optimal performance is removed, including personal idiosyncrasy and the joy of handiwork. For industrial culture, the number one goal is to satisfy needs quickly and without waste. Rather than lingering over a satisfying meal, the goal is to get customers in and out as quickly as possible. Why waste valuable time cooking a meal from scratch, when a frozen prepackaged pot pie is cheap and oh, so easy?

- *quantifiability*— In order to maximize efficiency, engineers calculate as much of the work process as possible: piece rates, materielle usage, worker movements, labor costs, recidivism rates. Things that cannot be calculated are devalued. Cost/benefit analyses weigh quality of life against cold, hard, calculable cash. The soul can't be weighed, so it must not exist.

  Quantifiability implies that more is better. The chain with the most stores must be the best. We buy, not the best-tasting sandwich, but the one with the largest pile of unidentifiable ground meat. Airlines advertise, not "We have the most pleasant flights," but "We fly to the most cities." The more you buy, the more you save!

- *predictability* — One of the major advantages of assembly line work is that the output of the assembly line is predictable. The phalanx of cars come marching off the assembly line, each exactly identical, with interchangeable parts, each with the same new car smell, the same ride, the same fluffy upholstery, the same engineering mistakes. Predictability substitutes for familiarity: wherever we go, the Days Inn is exactly the same, with the same cheerful desk clerks telling you to have a nice day, and the same style of insipid sit-com grinding out its laugh track from the satellite TV in your room. On your bus tour of Europe, there are no unpleasant surprises: 1 day per city, carefully sanitized local color, and the natives you meet all speak perfect English.

- *control* — Life (and in particular human behavior) is in many ways not inherently predictable, so the holy grail of predictability is only achievable through the hefty use of controls. Unpredictable humans are replaced and controlled by technology and bureaucracy: the ATM never miscounts, the computerized tram keeps its doors open for exactly 5.3 seconds, and the fast food worker does not get a chance to misspeak while regurgitating the manual's "Fries with that?" The customer can remain cost-effectively always right when s/he only has a choice of 5 menu items, and the high-intensity fluorescent lighting chases him or her out of the restaurant before s/he becomes an economic liability.

Each of these values certainly has its place. Inefficiency, incalculability, unpredictability, and lack of control are clearly not particularly preferable to their opposites. But Ritzer points out that under formal rationality, each of these values is elevated to an absolute. And when rationality is pushed too far, the result is, paradoxically, irrationality. A cheap fast food meal with huge portions, wolfed down in 5 minutes, is not necessarily better than a less 'efficient' home-cooked meal with quality ingredients. A packaged group tour with all activities carefully homogenized and isolated is safer, but not necessarily better, than a vacation requiring true contact with alien cultures and experiences. America has certainly pushed the envelope of homogenized, commodity-laden, safe, and predictable existence, but whether we truly maintain quality of life is an open question in a nation of the obese, who fuel the purchase of all the latest high tech fantasies by road raging across miles of asphalt from the suburbs to a 10 hour work day, then crawl back home to frozen dinners consumed silently in front of the stereo, big-screen high-definition TV. Even Taylorism, which subsumes all human values to the goal of efficiency, is inefficient in the sense that, by reducing work to repetitive motion, it wastes the worker's talents and judgment.

Under formal rationality, that which can be predicted and controlled is analyzed and rationalized. That which does not fit into those systems is ignored or denigrated. The result is the atomization, the fragmentation, the schizophrenization of daily life: the mindless suburban utopia as seen on TV masking violent death in the inner city; the back-to-nature marketing of enormous, environmentally destructive vehicles that will spend their lifespans only on urban highways and shuttling teenagers to and from the strip mall; taboos on sex mixing with advertising based largely on sex; unthinking bible-thumping TV evangelism providing hedges for the utter vacuity of spiritual values in public discourse. Little rationalities we surely have — the world's greatest can opener — but no sense as to how they should fit together into a meaningful life. Meaning itself — being one of those old-fashioned, inadequately calculable terms — no longer matters. Life, like our wetlands, is drained dry and replaced with a Wal-Mart.

> "In milling and baking, bread is deprived of any taste whatever and of all vitamins. Some of the vitamins are then added again (taste is provided by advertising). Quite similarly with all mass-produced articles. They can no more express the individual taste of producers than that of consumers. They become impersonal objects, however pseudo-personalized. Producers and consumers go through the mass production mill to come out homogenized and de-characterized — only it does not seem possible to reinject the individualities which have been ground out, the way the vitamins are added to enrich bread." ([Van den Haag, 1962], 183)

The law of industrialized culture is 'choose one of *n*.' Categories which were once abstract and qualitative are reified, making them quantitative and strictly delimited. Human qualities — your time, your work, your body — become commodities to be sold at will. Intelligence becomes IQ, family traits become genetic predispositions, class becomes income level, existential anxiety becomes a mental disease with its own number in the *Diagnostic and Statistical Manual of Mental Disorders*. In the process two things happen: definitions of these categories become so strict that they exclude much of what we find valuable in their informal counterparts; and in the process of setting up strict and delimited categories we lose the interrelationships between them. So, for example, in AI behavior is no longer a wholistic style of activity through which a being's existence is expressed; it becomes a set of atomically defined and separately written behaviors, of which industrialized agents choose one of *n*. Industrialization involves the loss of wholism and interconnections in favor of individually rationalized and atomically related parts; it leads to schizophrenia, the fragmentation of subjective experience itself.

Industrialized Science

The rise of industrialization has been accompanied by a rise in the importance of science and engineering. This link is not accidental. Science and technology give industrialists the ability to predict and control processes, providing the motive power for industrialization. At the same time, the industrialized world view is sympathetic to the scientific assumption that life is fundamentally a mechanical process that can be understood and controlled. Industrialization provides funding for the parts of science that are particularly useful for it, reinforcing those styles of science at the expense of others. Science in turn provides industrialization with

> "Machines — and machines alone — completely met the requirements of the new scientific method and point of view: they fulfilled the definition of 'reality' far more perfectly than living organisms. And once the mechanical world-picture was established, machines could thrive and multiply and dominate existence." ([Mumford, 1934], 51)

rationales for its activities. Science and industry become symbiotic, each reflecting aspects of the other.

Artificial Intelligence is no exception. From planning and scheduling of shop activities to robots for the assembly line, to reinforcement learning for process control, to automatic translation of manuals for equipment assembly, AI works on the problems of industrialization and, in turn, imbibes its values. Efficiency; quantifiability; predictability; control: Ritzer's values of industrialization are also the values of AI. They can be seen in the view of intelligence in AI, so different from most people's day-to-day experience of existence: the calls for rational, goal-seeking, provably correct agents, working efficiently to solve problems. They are reflected in the fundamental hope of AI: that most if not all of human behavior can be rationally analyzed, quantified, and reduced to algorithms reproducible in the machine.

AI is not alone; it represents in miniature the themes of post-industrial science, themes which are inherited from industrialization.

- *Reification* — Science works by approaching the multitude of phenomena of existence to find ways of sorting and categorizing them into well-defined categories. Animals are categorized into species, pain and discomfort into diseases, activity into behaviors. While the categories are always subject to revision, this involves the replacement of one kind of rigid definition with another, not the wholesale dissolution of categories. Classification is essential to science; without it, regularities cannot be discovered [Kirk and Kutchins, 1992].

- *Specialization* — Modern science has become more and more specialized and esoteric. Science is split into many heterogeneous sciences, each studying its own phenomena in its own way. It is not even clear how to relate the subfields of a particular discipline, let alone how biology, chemistry, physics, psychology, and sociology can be combined to form one consistent world view. In this sense, science, like modern consciousness, is fragmented and incoherent.

- *Atomization* — The methods of science involve breaking up phenomena into subparts, studying these parts in isolation, and trying to reconstruct the full phenomena from these presumably independent parts. "[T]he ideology of modern science... makes the atom or individual the causal source of all the properties of larger collections. It prescribes a way of studying the world, which is to cut it up into the individual bits that cause it and to study the properties of these isolated bits" ([Lewontin, 1991], 12-13). Lewontin points out that this way of conceptualizing the world, which comes to us in post-industrial society so naturally, would have been unthinkable in the Middle Ages, when nature was seen as essentially wholistic; dissecting nature was thought to destroy its essence. But when all of society is thought to consist of atomic, free, and independent individuals, it is not so strange to think of nature this way, too.

- *Standardization* — Science understands all electrons, all anxiety disorders, all elephants as basically alike. Yes, there are individual differences within a category, but the very construction of a category implies for scientists that the rules regarding that category apply to its members in the same way. Under industrialization, "the human qualities and idiosyncrasies of the worker appear increasingly as *mere sources of error* when contrasted with these abstract special laws functioning according to rational predictions"([Lukács, 1971], 89). Similarly, any variation of the behavior of scientifically classified objects from the norm is considered statistically manageable 'noise.'

- *Formalization* — Science differs from alchemy in that the individual, material, idiosyncratic attributes of objects are considered unimportant. Rather, the ultimate goal of science is the reduction of the material to mathematics. Truly elegant scientific theories represent complex reality in terms of a few simple, well-defined laws, formal representations into which scientific objects can be plugged with the minimal possible reference to their idiosyncratic individuality.

  For the same reason, the context of scientific work is often minimized or forgotten. The scientist reduces not only the idiosyncrasies of the scientific object, but tries to remove his or her own idiosyncrasies as individual from the results of scientific work. "[S]cientific experiment is contemplation at its purest. The experimenter creates an artificial, abstract milieu in order to be able to *observe* undisturbed the untrammelled workings of the laws under examination, eliminating all irrational factors both of the subject and the object. He strives as far as possible to reduce the material substratum of his observation to the purely rational 'product', to the 'intelligible matter' of mathematics" ([Lukács, 1971], 132).

- *Mechanization* — The scientific worldview is a mechanical worldview. References to the 'soul,' to God, to the unknowable, to the very possibility of free will, which might be considered signs of a healthy respect for the limits of human ways of knowing, instead are considered highly suspect and even laughable. Instead, one of the ultimate goals of scientific knowledge is the synthesis of physics, biology, and psychology, into a complete description of human beings as fully mechanical systems. The body is a machine; with the development of cognitive science, the mind is a machine as well. This mechanistic viewpoint is not seen as metaphor, but as reality: Lewontin notes that while in Descartes' day, the world was considered to be *like* a machine, in our post-industrial existence we really consider the world to *be* a machine [Lewontin, 1991].

Post-industrial science works on the theory of the assembly line: "the process as a whole is examined objectively, in itself, that is to say, without regard to the question of its execution by human hands, it is analysed into its constituent phases; and the problem, how to execute each detail process, and bind them all into a whole, is solved by the aid of machines, chemistry, &c." ([Marx, 1967], 380). Like industrial engineering, science understands life by decontextualizing and dissecting it — taking it apart, analyzing each part separately, and then combining these independent forms of understanding into a functional but nevertheless fragmented whole. "Scientific... 'good sense' operates in essentially the same way as common sense: isolation of the typical individual (considered outside the real flow of its actions; as essentially dead); decomposition into parts and determination of intrinsic qualities (dissection); logical recomposition into an organic whole exhibiting signs of 'life' (artificial resuscitation)..." ([Massumi, 1992], 97). The wholism of science is a summation of individual, independent parts, each rationalized separately, each acting without reference to the others: Lukács's "objective synthesis" (88), the sum of calculation, arbitrarily connected.

The result of this process of fragmentation is schizophrenia. The object of study in science is split into a thousand pieces, each of which is rationalized separately and reunited in a parody of wholism. The union of these parts is incoherent; they may fit together in places, but only by accident; their necessary connections were left behind at the moment of dissection. And all that is not amenable to rational analysis is also left behind, forming a residue of noise that marks the limit-point of rationality. Schizophrenia is the uncategorizable; in the feedback loop between rationality and incoherence, schizophrenia is the short-circuit.

Case Study 2: AI as Institutionalization

"A man who says that men are machines may be a great scientist. A man who says he *is* a machine is 'depersonalized' in psychiatric jargon." ([Laing, 1960], 12)

So far, schizophrenia has functioned as an abstract term in this thesis, a breakdown in overall cohesion that comes about when life is micro-rationalized. However, schizophrenia is not simply a trendy theoretical term, but also a lived reality; and there are important relationships between AI as an intellectual discipline and the experience of being a schizophrenic person, especially as understood by institutional psychiatry.

In particular, the flip side of the AI concept of consciousness-as-machine is the schizophrenic experience of self-as-machine. Critics of institutional psychiatry argue that this 'delusion' (or, better put, unique and painful existential position) is reinforced under a scientific psychiatry that attempts to explain schizophrenia in mechanistic terms. Taking an objective perspective on schizophrenia, seeing patients' behavior not as an expression of their unique selves but as mere symptomatology of a disease, fundamentally involves denying those patients' already marginal experience of personhood, rendering schizophrenics incomprehensible, their speech no more than word salad. Institutional psychiatry, by objectivizing the schizophrenic and schizophrenia, splits the schizophrenic from his or her context and from his or her disease, repeating the fragmentation of subjective experience that is a hallmark of schizophrenia. These moves parallel the decontextualization, reification, and fragmentation of behavior that occurs in AI. In this section, we will look at these problems in institutional psychiatry in more detail; proposed solutions to the problems inherent in this mechanization of patient psychology will become the basis for rethinking AI in Chapter 4.

Institutionalization as Mechanization

"Toute l'histoire de la folie... n'est que la description de l'automatisme psychologique." ([Janet, 1889], 478)

In the late 1800's, Pierre Janet identified one of the more baffling symptoms of schizophrenia — the *sentimente d'automatisme*, or subjective experience of being a machine. This feeling is the flip side of AI's hoped-for machinic experience of being subjective, and is described by one patient this way: " 'I am unable to give an account of what I really do, everything is mechanical in me and is done unconsciously. I am nothing but a machine' " (an anonymous schizophrenic patient; cited in ([Ronell, 1989], 118)). R. D. Laing describes how some schizophrenic patients experience or fear experiencing themselves as things, as its, instead of as people [Laing, 1960]. Schizophrenia is, for some, a frightening feeling of being drained of life, of being reduced to a robot or automaton.

This feeling of mechanicity is correlated with a fragmentation of the affected patient's being; sometimes, a schizophrenic patient's very subjectivity seems to be split apart.

> In listening to Julie, it was often as though one were doing group psychotherapy with the one patient. Thus I was confronted with a babble or jumble of quite disparate attitudes, feelings, expressions of impulse. The patient's intonations, gestures, mannerisms, changed their character from moment to moment. One may begin to recognize patches of speech, or fragments of behaviour cropping up at different times, which seem to belong together by reason of similarities of the intonation, the vocabulary, syntax, the preoccupations in the utterance or to cohere as behaviour by reason of certain stereotyped gestures or mannerisms. It seemed therefore that one was in the presence of various fragments, or incomplete elements, of different 'personalities' in operation at the one time. Her 'word-salad' seemed to be the result of a number of quasi-autonomous partial systems striving to give expression to themselves out of the same mouth at the same time. ([Laing, 1960], 195-6)

Laing goes on to describe Julie's existence in ways that are eerily similar to the problems with autonomous agents we discussed in the last chapter — all the more eery because we are talking about actual, painful human experience and not a theoretical description of a machine: "Julie's being as a chronic schizophrenic was... characterized by lack of unity and by division into what might variously be called partial 'assemblies', complexes, partial systems, or 'internal objects'. Each of these partial systems had recognizable features and distinctive ways of its own" (197). Like the parts of behavior-based agents, each subsystem exists independently, with its own perception and action. Subsystems communicate, in Brooks' phraseology, 'through the world,' not by being integrated as a unified whole:

> Each partial system seemed to have within it its own focus or centre of awareness: it had its own very limited memory schemata and limited ways of structuring percepts; its own quasi-autonomous drives or component drives; its own tendency to preserve its autonomy, and special dangers which threatened its autonomy. She would refer to these diverse aspects as 'he', or 'she', or address them as 'you'. That is, instead of having a reflective awareness of those aspects of herself, 'she' would *perceive* the operation of a partial

> system as though it was not of 'her', but belonged outside. (198).[3]

While we can presume that artificial systems do not particularly care about being fragmented, for schizophrenic patients this feeling of coming apart, of losing life, of being reduced to a machine, is intensely painful. It is therefore ironic that psychiatric institutions themselves reinforce this feeling of mechanicity and lack of autonomous self. Erving Goffman, in his anthropological study *Asylums* [Goffman, 1961], analyzes psychiatric institutions, and concludes that one of their features is the attempted mechanization of their inmates.

Goffman's interest is in 'total institutions' such as psychiatric institutions, jails, and concentration camps, i.e. institutions that are barricaded from the rest of society and encompass all of their inmates' lives. From the beginning of an inmate's stay at such an institution, s/he is asked to give up his or her own identity in order to make for smoother processing by institutional bureaucracy.

> Admission procedures might better be called 'trimming' or 'programming' because in thus being squared away the new arrival allows himself to be shaped and coded into an object that can be fed into the administrative machinery of the establishment, to be worked on smoothly by routine operations. Many of these procedures depend upon attributes such as weight or fingerprints that the individual possesses merely because he is a member of the largest and most abstract of social categories, that of human being. Action taken on the basis of such attributes necessarily ignores most of his previous bases of self-identification. (16)

The admission procedures mark the beginning of a period of standardization, where inmates' individual identity is denied. "The admission procedure can be characterized as a leaving off and a taking on, with the midpoint marked by physical nakedness. Leaving off of course entails a dispossession of property, important because persons invest self feelings in their possessions. Perhaps the most significant of these possessions is not physical at all, one's full name; whatever one is thereafter called, loss of one's name can be a great curtailment of the self" (17). In place of patients' initial feeling of individuality, the institution enforces a homogeneous, standardized life, a homogeneity that is reflected in patients' environments, including their physical environment and clothing. "Once the inmate is stripped of his possessions, at least some replacements must be made by the establishment, but these take the form of standard issue, uniform in character and uniformly distributed. These substitute possessions are clearly marked as really belonging to the institution and in some cases are recalled at regular intervals to be, as it were, disinfected of identifications" (19).

The institutions' push to standardization, not only of patients' appearance, but of patients' very existence, is seen in the continuous intimate

[3] This splitting into subsystems is not the same thing as multiple personality. They are not experienced as completely separate individuals. In addition, Laing posits the subsystems as an explanatory mechanism that makes Julie's utterances more understandable; no one can directly know Julie's subjective experience, and she is not in a position to articulate it.

regulation of patients' lives: "[T]he inmate's life is penetrated by constant sanctioning interaction from above, especially during the initial period of stay before the inmate accepts the regulations unthinkingly.... The autonomy of the act itself is violated" (38). The nature of institutionalization is to (further) reduce patients' individuality and sense of autonomy. Patients must constantly ask for permission to do anything other than what the institution has planned for them; often times, even these requests are ignored, since patients may not be considered worth listening to. "Of course you had what they called an [sic] hearing but they didn't really want to hear you" ([Washington, 1991], 50). Over time, all resistance is worn down until patients passively accept the institution's decisions for them, becoming, at least in the eyes of its staff, little more than bureaucratic objects to be pushed and pulled into place.

Institutional Impoverishment of Meaning

So far, the mechanization of the inmate is similar in all total institutions. But psychiatric institutions are unique in that everything patients do — the last remaining bastion of individual expression — is treated as merely symptomatic. Patients are constantly monitored, their behavior continuously being examined for signs of illness.

> All of the patient's actions, feelings, and thoughts — past, present and predicted — are officially usable by the therapist in diagnosis and prescription.... None of a patient's business, then, is none of the psychiatrist's business; nothing ought to be held back from the psychiatrist as irrelevant to his job. (358)

In our everyday lives, we expect our utterances to be understood at face value; we become angry if, instead of trying to understand what we are saying, someone merely interprets it: "You are only so angry because you are still hypersensitive about your mother abandoning you." But in the institution, patients' words and actions are often simply interpreted as signs of illness. Rather than acting, patients *signify*. The patient's actions only function insofar as they are informational — they only *act* as ciphers, which it is then the responsibility and right of the doctor to decode. As a cipher, a patient's words can never be taken seriously as such; rather than being understood to refer to their intended meaning, the words are used to place the patient in the narrative of the doctor's diagnosis. "When you spoke, they judged your words as a delusion to confirm their concepts" ([Robear Jr., 1991], 19). The patient's acts are robbed of meaning so that another system of meaning can be imposed.

Maurice Blanchot expresses the frustrating abandonment of identity in this situation:

> I liked the doctors quite well, and I did not feel belittled by their doubts. The annoying thing was that their authority loomed larger by the hour. One is not aware of it, but these men are kings. Throwing open my rooms, they would say, 'Everything here belongs to us.' They would fall upon scraps of thought: 'This is ours.' They would challenge my story: 'Talk,' and my story would put itself at their service. In

> haste, I would rid myself of myself. I distributed my blood, my innermost being among them, lent them the universe, gave them the day. Right before their eyes, though they were not at all startled, I became a drop of water, a spot of ink. ([Blanchot, 1981], 14)

"Automatization eats away at things, at clothes, at furniture, at our wives, and at our fear of war." ([Shklovsky, 1990], 5)

The patient, rather than being treated as a full human being, is seen as a sign or symbol. Victor Shklovsky calls this reduction of a complex individual to a simple sign — a move which also occurs in AI when we reduce a complex behavior to a simple atom like 'hunting' or 'eating' — "automatization" [Shklovsky, 1990]. He argues that automatization causes one to forget the full richness of the actual object of automatization, replacing it with a single word. Similarly, by reducing patients to a set of signs to be interpreted, the institution only recognizes a small part of them.

The difficulty is that, once the bureaucratic system has standardized the patient, and the psychiatric system has ignored what the patient tries to say and do in favor of a symptomatic view, there is a huge gap between the the institution's mechanized view of the patient as symbol and the patient's experience of him- or herself as an individual person. The patient as a complete, subjective, and unique individual is simply not understandable under the rubric of the psychiatric institution. In this sense, the patient becomes invisible.

"I had been asked: Tell us '*just* exactly' what happened. A story?... I told them the whole story and they listened, it seems to me, with interest, at least in the beginning. But the end was a surprise to all of us. 'That was the beginning,' they said. 'Now get down to the facts.' How so? The story was over!" ([Blanchot, 1981], 18)

> The whole of me passed in full view before them, and when at last nothing was present but my perfect nothingness and there was nothing more to see, they ceased to see me too. Very irritated, they stood up and cried out, 'All right,where are you? Where are you hiding? Hiding is forbidden, it is an offense,' etc." ([Blanchot, 1981], 14)

The patient as understood by the institution is reified, atomized, mechanized, standardized, formalized, reduced to a mere ghost of his or her internally experienced self. Understood symptomatically, the patient's subjective experience is ignored. Susan Baur describes this limitation of the institutional approach to mental illness:

> I... believe that the medical model of mental illness excludes too much of the patient. Using this model, only parts of the patient are considered, and even when these parts are assembled by a multidisciplinary team into a manikin of a schizophrenic or of a manic-depressive, the spirit that animates the real person gets lost. Especially in chronic cases where mental illness and the desperately clever adaptations it inspires have become central to an individual's personality, the patient's own story and explanations — his delusions and imaginary worlds — must be included ([Baur, 1991], 105-6).

This leaves patients, sadly, ununderstood by the very institutions which are supposed to house and heal them.

Institutionalized Science

The fundamental problem of institutionalization is the bureaucratization of the patient. In bureaucracies, 'understanding' is reduced to categorization: instead of seeing each person as a unique, complex individual, people entering bureaucracies are classified and operated on according to standard, objective categories. Not many of those who read this thesis have been institutionalized; but all of us can recognize the feeling of frustration and alienation that comes from being treated as a thing by a bureaucrat.

The difficulty for institutional psychiatry is that, by treating the patient as a set of signs to be interpreted, the 'real' patient is left behind. The patient is formalized, reduced to a set of somewhat arbitrarily connected symptoms. Institutional psychiatry leaves the living patient out when it takes that formalized image of the patient for the patient him- or herself. The patient is no longer a living, unique, complex individual, but fragmented into a pile of signs: "she is autistic," "she shows signs of depersonalization," "she lacks affect."

This move — and the problems it brings — are paralleled in modern science. In science, the material, idiosyncratic properties of the objects to be studied are reduced to formal theories, preferably stated in terms of mathematics. While there is nothing wrong with formalization per se, difficulties come about when, as Katherine Hayles describes, the formalized theory is seen as more real than — or even *causing* — the material things being described [Hayles,]. One example of this is Dawkins's theory of the "selfish gene:" starting from theories of evolution, Dawkins argues that humans are 'really' no more than large bags of flesh whose only purpose is the propagation of genetic information [Dawkins, 1989] — thereby belittling the importance of the life history of individual living beings, which is only partly determined by genes. The same move is made in the institution: the patient is seen as fundamentally fragmented and symptomatic, structured as in psychiatric theory, not as a complex, embodied human being; his or her behavior is caused not by the patient's will but by a disease. According to Hayles, the information sciences sometimes make the same mistake: they see the world as 'really' a flow of information, with its materiality and noisy complexity an accidental after-effect.

Humanists will recognize reductionism.

In each of these areas, the wholistic and not-entirely-comprehensible aspects of the studied phenomena are forgotten, set aside in favor of a simpler and more elegant theory. But if your goal is to understand and engage in real life — or, in the case of AI, to be able to generate creatures that are in some sense truly alive — then it is best not to become too enamoured of your *theories* of life. If the only view of life you value is formalized and rationalized knowledge, then the world, which is probably neither formal nor rational, will always exceed it. The world is wholistic, complex, incompletely knowable; if only fragmented, elegant, and complete theories of the world are allowed, the actual world will seem to be incomprehensibly heterogeneous: schizophrenic. In this sense, schizophrenia in science is a result of the fragmentation that clean categorization brings about; it represents the limits of categorical knowledge.

Science and Alienation

So far, we have looked at science in terms of its relations to industrialization and to institutionalization. Industrialized science repeats the themes of assembly line work, using the processes of reification, specialization, atomization, standardization, formalization, and mechanization. Institutionalized science follows institutional psychiatry in reducing the objects of its study to formalized, fragmented versions of them. When both cases are taken to extremes, science loses something important: the subjective, idiosyncratic, incompletely knowable aspects of what it studies; the 'meaning' in life.

This is because industrialization and institutionalization share a negative side-effect: alienation. The term 'alienation' is used in multiple ways, but it can be fundamentally understood as a subjective feeling of being cut off: cut off from oneself, cut off from others, cut off from one's own actions. Under industrialization, workers are said to suffer from alienation because they are separated from the results of their work; instead of acting directly on their products, they merely tend machinery. Under institutionalization, the patient is alienated from the role which s/he is expected to play, and with which s/he may only marginally identify. Alienation is a fragmentation of life, a draining-away of the meaning of life, as the parts of one's life — one's self, one's friends, one's work — become separated, each part functioning atomically, with no subjective feeling of interconnection or wholism.

Modern science, too, is alienating. Unfortunately, the goal of reliable knowledge in science is often understood as necessitating a split between the individual scientist and the things or people which s/he studies, i.e. a subject / object divide. Science is generally understood not as a result of individual lives expressing themselves within a community of shared traditions, but as a self-contained, self-propelling force with its own logic, somehow only incidentally involving human beings. Even the very use of the word "I" in scientific papers is considered suspect; "the author is advised to avoid the use of first-person pronouns," as an otherwise extraordinarily helpful anonymous reviewer report of one of my papers elegantly circumlocuted it. "The experiment was conducted," "Results showed that... ," "It was noticed that... ," you read in the literature, as though research happened by itself, and the scientist only stopped by the office once a week to pick up the finished paper.

The scientist him- or herself is alienated in the sense that the product of his or her work — science itself — tends to feel independent of the scientist's personal existence. Indeed, the argument is often made, particularly in the natural sciences, that the individual scientist does not really matter; if a particular scientist had not done a certain piece of work, someone else would have done it. But in addition to the scientist itself, the very things that science studies are also alienated: they are atomized, fragmented, dissected, both literally and metaphorically; the very term 'science' probably comes from the Latin 'scindere,' to split [Gove, 1986].[4]

To be precise, it is unlikely that albino mice in a medical experiment have a subjective experience of alienation. But alienation can certainly

[4] Thanks to Stefan Helmreich for this observation.

be the experience of humans who try to understand themselves through the lens of science. Try being hospitalized for an unknown disorder, and watch the specialists turn you one by one into a skeletal system, a gastro-intestinal system, a nervous system, an immune system, and, if it turns out your problem is wholistic, a hysteric. Try to understand what makes you tick by reading the latest results in experimental psychology and statistical sociology; the more studies you read, the more multiple you feel, the less you are able to synthesize them into a coherent worldview. Seen through the lens of science, you are split into biology, psychology, and sociology, and in each of these realms into a thousand more subfields and experimental results. Good luck finding yourself!

Alienation is bad for science because it makes the things science studies seem fragmented. Science breaks things into pieces to study them; whether or not they 'actually' are fragmented (probably not), they end up looking that way to us. This means that the results of science can be misleading. In this section, we will look at several ways of doing science that try to resolve the problems of objectivist science. First, we will look again at schizophrenia — now understood in psychiatric terms — to understand concretely in this example how objectivist science, in alienating doctor from patient, can unconsciously fragment the patient, rendering him or her unnecessarily incomprehensible. We will then look at an alternative approach proposed by anti-psychiatry to find alternative ways of doing science that avoid the pitfalls of alienation.

Alienation and Objectivist Science: The Divided Self

Earlier, we noted that schizophrenia sometimes includes an alienation-from-self in that the self is experienced as split into different parts. R. D. Laing describes schizophrenia as including, not just a division within the parts of the self, but also a disruption between the self and the rest of the world.

> The term schizoid refers to an individual the totality of whose experience is split in two main ways: in the first place, there is a rent in his relation with his world and, in the second, there is a disruption of his relation with himself. Such a person is not able to experience himself 'together with' others or 'at home in' the world, but, on the contrary, he experiences himself in despairing aloneness and isolation; moreover, he does not experience himself as a complete person but rather as 'split' in various ways, perhaps as a mind more or less tenuously linked to a body, as two or more selves, and so on. ([Laing, 1960], 17)

"The standard texts contain the descriptions of the behaviour of people in a behavioural field that includes the psychiatrist. The behaviour of the patient is to some extent a function of the behaviour of the psychiatrist in the same behavioural field. The standard psychiatric patient is a function of the standard psychiatrist, and of the standard mental hospital. The figured base, as it were, which underscores all Bleuler's great description of schizophrenics is his remark that when all is said and done they were stranger to him than the birds in his garden." ([Laing, 1960], 28)

Laing describes how schizophrenics may construct a 'false-self' system, through which they present a false front to the world, while keeping their self-identified 'real' selves safely hidden away. This false-self mechanism may then be partly responsible for a patient's further deterioration; without the confirmation of self that social interaction brings, patients' real selves are in danger of wasting away.

This split between a schizophrenic and their surrounding environment has been more generally noted. Schizophrenic language itself may lack reference to context; a patient may, for example, be laughing while recounting a heart-rending story [American Psychiatric Association, 1994]. Social withdrawal or a 'break with reality' is also common. "The person may be so withdrawn from the world that h/she is absorbed entirely in his/her mixed-up thoughts" ([Webb *et al.*, 1981], 72).

Understanding schizophrenics can therefore be difficult because it is often hard to establish social contact with them. Because schizophrenics may fear true social contact, they may even actively undermine the doctor's understanding. "A good deal of schizophrenia is simply nonsense, red-herring speech, prolonged filibustering to throw dangerous people off the scent, to create boredom and futility in others. The schizophrenic is often making a fool of himself and the doctor" ([Laing, 1960], 164). This complicated the doctor's job; it is simply hard to understand someone who refuses to interact.

For psychiatrists like Laing, one of the main avenues toward understanding schizophrenia, then, is to break down the barrier between schizophrenic patients and their social worlds by engaging in personal relationships with them, i.e. by putting patients back into their social contexts. But Laing finds that the methods and language of clinical psychiatry actually undermine his goal to connect with the patient as a human being. This is because, rather than treating the patient as a person, psychiatrists see patients as a bundle of symptomatology. Mechanistic explanations reduce the patient to a bundle of pathological processes.

This 'clinical detachment,' by which the patient can be seen as a mere instance of a disease, is considered good because treating the person as a whole person would mean entering into a personal relationship with them, undermining objectivity.

> [T]here is a common illusion that one somehow increases one's understanding of a person if one can translate a personal understanding of him into the impersonal terms of a sequence or system of *it*-processes. Even in the absence of theoretical justifications, there remains a tendency to translate our personal experience of the other as a person into an account of him that is depersonalized. (22)

But just as it is inaccurate to describe an animal or object in anthropomorphic terms, it is equally inaccurate to picture a human as an animal or automaton.

"As a psychiatrist, I run into a major difficulty at the outset: how can I go straight to the patients if the psychiatric words at my disposal keep the patient at a distance from me? How can one demonstrate the general human relevance and significance of the patient's condition if the words one has to use are specifically designed to isolate and circumscribe the meaning of the patient's life to a particular clinical entity?" [Laing, 1960], 17)

Fundamentally, the stumbling block for objectivist psychiatry is that a detached, impersonal attitude does not lead to a view of the patient independent of the psychiatrist's personal attitudes. This is because the objective, clinical approach that psychiatrists may take is itself part of the schizophrenic patient's situation. The 'objectivity' the psychiatrist takes on itself influences what the patient does and how the psychiatrist can come to understand him or her.

> The clinical psychiatrist, wishing to be more 'scientific' or 'objective', may propose to confine himself to the 'objectively' observable behaviour of the patient before him. The simplest reply to this is that it is impossible. To see 'signs' of 'disease' is not to see neutrally.... We cannot help but see the person in one way or other and place our constructions

> or interpretations on 'his' behaviour, as soon as we are in a relationship with him. (31)

Even the objectivist psychiatrist is constructing a particular kind of relationship with the patient, one that cuts off the possibility of human understanding. By treating the patient as separate, as not a person, as a thing, the patient as human is rendered incomprehensible.

Laing argues that institutional psychiatric practice cannot fully understand schizophrenia because it actually *mimics* schizophrenic ways of thinking, depersonalizing and fragmenting patients. "The most serious objection to the technical vocabulary currently used to describe psychiatric patients is that it consists of words which split man up verbally in a way which is analogous to the existential splits we have to describe here" (19). Clinical language atomizes and reifies patients, studying them in isolation from their worlds and from the psychiatrist.

> Unless we begin with the concept of man in relation to other men and from the beginning 'in' a world, and unless we realize that man does not exist without 'his' world nor can his world exist without him, we are condemned to start our study of schizoid and schizophrenic people with a verbal and conceptual splitting that matches the split up of the totality of the schizoid being-in-the-world. Moreover, the secondary verbal and conceptual task of reintegrating the various bits and pieces will parallel the despairing efforts of the schizophrenic to put his disintegrated self and world together again. (19-20)

By studying schizophrenics in isolation and in parts, psychiatry threatens to itself become schizophrenic, and schizophrenics incomprehensible.

Anti-Psychiatry: Science in Context

If objectivist psychiatry distorts and fragments schizophrenia, rendering it incomprehensible, are there other ways of doing science that avoid alienation? Laing and other sympathetic colleagues in the 60's and 70's, termed *anti-psychiatrists* for their opposition to mainstream psychiatry, suggest that the schizophrenizing aspects of institutional psychiatry can be avoided by understanding schizophrenia in the context of the patient's life. If schizophrenia is to be understood, anti-psychiatrists argue, we need to think of schizophrenics, not as self-contained clusters of symptoms, but as complex humans. This means studying them, not in a vacuum, but in relation to both their lifeworlds and to the people who study and treat them, including psychiatrists themselves.

"A schizophrenic out for a walk is a better model than a neurotic lying on the analyst's couch. A breath of fresh air, a relationship with the outside world." ([Deleuze and Guattari, 1977], 2)

The difference between these approaches can be understood by contrasting objectivist and subjectivist descriptions of patient behavior. The clinical approach reifies the patient's behavior into a cluster of pathological symptoms, with no apparent relation to each other or the patient's broader life experience:

> [S]he had auditory hallucinations and was depersonalized; showed signs of catatonia; exhibited affective impoverishment and autistic withdrawal. Occasionally she was held to be 'impulsive.' ([Laing and Esterson, 1970], 32)

The phenomenological approach advocated by anti-psychiatrists, on the other hand, tries to understand the patient's experience of herself as a person:

> [S]he experienced herself as a machine, rather than as a person: she lacked a sense of her motives, agency and intentions belonging together: she was very confused about her autonomous identity. She felt it necessary to move and speak with studious and scrupulous correctness. She sometimes felt that her thoughts were controlled by others, and she said that not she but her 'voices' often did her thinking.

Anti-psychiatrists believe that statistics and symptomatology, the foundations of institutional psychiatry, are misleading because they reduce the patient to a mass of unrelated signs. Instead of leading to a greater understanding of the patient, the patient's subjective experiences are lost under a pile of unconnected data.

> It is just possible to have a thorough knowledge of what has been discovered about the hereditary or familial incidence of manic-depressive psychosis or schizophrenia, to have a facility in recognizing schizoid 'ego distortion' and schizophrenic ego defects, plus the various 'disorders' of thought, memory, perceptions, etc., to know, in fact, just about everything that can be known about the psychopathology of schizophrenia or of schizophrenia as a disease without being able to understand one single schizophrenic. Such data are all ways of *not* understanding him. ([Laing, 1960], 33)

Instead of trying to extract objectively verifiable data about the patient, anti-psychiatrists believe psychiatry should be based on *hermeneutics*, a subjective process of interpretation which aims for a better understanding of the way in which the schizophrenic patient experiences life. Laing finds that when schizophrenic patients are treated 'subjectively' — that is to say, when attempts are made, not to catalog their symptoms, but to understand their phenomenological viewpoints, even when they include such apparently alien components as delusions or hallucinations — schizophrenia can be made much more comprehensible. In *Sanity, Madness, and the Family*, Laing and Esterson give 11 case studies of schizophrenic patients whose behavior, initially incomprehensible and even frightening, is made understandable by putting it in the context of the patient's family life. For example, a patient with a delusion that other people are controlling her thoughts is found to live in a family where her parents undermine every expression of independent thought, telling her that they know better than her what she thinks.

It is important to note that understanding a schizophrenic patient is not the same as curing him or her. Giving meaning to delusions and hallucinations does not take them away or reduce their effect on a patient's life. Nevertheless, complementing clinical understanding of a patient with phenomenological interpretation of the patient's life-world gives a fuller picture of the patient as human being and provides better understanding of the nature of schizophrenia in this individual person.

Anti-psychiatrists believe that the concept of schizophrenia as a pathological disorder affecting individuals in isolation is misleading. When studied in context, schizophrenic symptomatology that otherwise seems bizarre and inexplicable starts to make sense; in this sense, schizophrenia is a sane response to an insane situation. Anti-psychiatrists note that schizophrenic patients are sometimes the locus of negative tension in their families; they hypothesize that patients may take on the label of 'sick' so that their families can avoid introspection into the negative aspects of their psychodynamics. In addition, cultural influences — the broader atomization and depersonalization of post-industrial life — may itself be 'schizophrenizing,' a factor which is forgotten when research focuses on the sole, sick individual instead of the society that in some sense causes his or her illness. Finally, the very reification of schizophrenia as a disease an individual 'has' is misleading, because it separates a patient from his or her behavior and pathologizes it.

In essence, anti-psychiatrists make not only an epistemological argument, but an ethical one. According to anti-psychiatrists, the use of schizophrenia in institutional psychiatry is not only incorrect, but morally wrong. Treating people as objects not only leaves them incomprehensible in their humanity; it also makes it easier to treat them as objects, cogs in the institutional machine. Depersonalization is not only an intellectual viewpoint, but the daily experience of institutionalized patients, which ranges from mild annoyances in exclusive, private wards, to the warehousing of humanity in large, public institutions, to the absolutely inhuman conditions of institutes for the criminally insane (see e.g. [Viscott, 1972]). "We are a special breed of farmyard animals," as Sylvain Lecocq wrote his doctor a year and a half before hanging himself from his hospital bed ([Lecocq, 1991], 160).

Anti-psychiatrists often antagonize more mainstream psychiatrists, in much the same way that the cultural critics of science antagonize scientists. In essence, the anti-psychiatrists argue that schizophrenia is a social construct, supported by the medical and institutional establishments, but not necessarily particularly helpful in treating those considered mentally ill. Psychiatrists interpret this as an argument that schizophrenia is a fiction, a mere social label, and that objectivist psychiatry is, in essence, colluding with families to label otherwise perfectly healthy people as dysfunctional. And some anti-psychiatrists basically agree with this perception:

> [S]chizophrenia is a micro-social crisis situation in which the acts and experiences of a certain person are invalidated by others for certain intelligible cultural and micro-cultural (usually familial) reasons, to the point where he is elected and identified as being 'mentally ill' in a certain way, and is then confirmed (by a specifiable but highly arbitrary labelling process) in the identity 'schizophrenic patient' by medical or quasi-medical agents. ([Cooper, 1967], 2, emphasized in original)

One of the results of this mutual antagonism is a backlash in institutional psychiatry, as psychiatrists attempt to disprove the unattractive claims of anti-psychiatry by showing that schizophrenia is basically a biological illness which can be objectively identified. Anti-psychiatry was dealt another blow in the 80's, when its demonization of institutionalization was used as a pretext for the economically attractive Reagan-era depopulation and closure of mental hospitals. The former inmates, now dumped on the streets basically untreated and unable to cope with life, can be seen in most major American urban centers, an apparent living testament to anti-psychiatry's bankruptcy — if one ignores the fact that anti-psychiatrists never proposed getting rid of the problems of institutions by simply kicking all the patients out.

But even the *Diagnostic and Statistical Manual of Mental Disorders*, which represents the conservative mainstream of psychiatry, notes the sociocultural face of schizophrenia: that, for example, schizophrenia is more prevalent and harder to treat in industrialized nations [American Psychiatric Association, 1994]. Schizophrenia is probably not *merely* a social label in the way more extreme anti-psychiatrists seem to imply — it is, for example, more prevalent among relatives of already-diagnosed schizophrenics, even when raised apart. But, at the same time, schizophrenia is also clearly influenced by environmental factors (for example, it is not unusual for only one half of a monozygotic twin to have it). Schizophrenia clearly *does* depend on the sociocultural context within which the labeled schizophrenic lives. The anti-psychiatric interest in contextualization therefore lives on, even in mainstream psychiatry.

Alternatives to Alienated Science

Anti-psychiatry rejects the objectivist stand of institutional psychiatry, arguing that understanding human beings is qualitatively different from understanding inanimate objects as in physics.

> It may be maintained that one cannot be scientific without retaining one's 'objectivity.' A genuine science of personal existence must attempt to be as unbiased as possible. Physics and the other sciences of things must accord the science of persons the right to be unbiased in a way that is true to its own field of study. If it is held that to be unbiased one should be 'objective' in the sense of depersonalizing the person who is the 'object' of our study, any temptation to do this under the impression that one is thereby being scientific must be rigorously resisted. Depersonalization in a theory that is intended to be a theory of persons is as false as schizoid depersonalization of others and is no less ultimately an intentional act. Although conducted in the name of science, such reification yields false 'knowledge'. It is just as pathetic a fallacy as the false personalization of things. ([Laing, 1960], 24)

The belief in objectivity — in the sense of belief that the psychiatrist as a knowing subject can be cleanly divided from the patient, who is an object to be understood mechanically — fundamentally distorts our perception of patients, simply because patients are always already in a human relationship with the doctor, even when that relationship consists of the doctor ignoring the patient's humanity.

Anti-psychiatrists not only criticize objectivist science as alienated and alienating; they also develop new ways of achieving the goals of psychiatry that do not have the same flaws. Anti-psychiatry argues that symptomatic views of mental patients actually reinforce schizophrenia by depersonalizing patients, fragmenting them, and removing them both physically and epistemologically from their contexts. Instead, anti-psychiatrists develop a new practice, one that is based on respect for the patient as a complete person and attempts to interpret his or her behavior not in isolation but in the context of the patient's complete lifeworld.

This context is not just limited to the patient's family. The really novel step the anti-psychiatrists take is to become aware of the role *they themselves* play in interpreting and interacting with the patient. Anti-psychiatrists do not see themselves as looking on the patient's life from the outside; they understand that even as they are trying to study the patient in as unbiased a way as possible, they cannot help but be in a human relationship with the schizophrenic that effects how they come to understand the patient him- or herself.

The fundamental recommendation anti-psychiatry makes for the methodology of psychiatry is this: *the patient should be studied in context.* This means on the one hand that the 'parts' of the patient — his or her symptoms, 'subsystems,' actions, language – should be studied in relation to one another, forming a unified rather than fragmentary picture of the patient as a person. On the other hand, it means that the patient should be studied in a social context, a context which includes the people who are judging him or her.

This proposal for addressing the problems of alienated science is similar to ones that have been raised in other fields. In neurology, for example, Kurt Goldstein argues that the fragmentation of organisms as necessarily occurs in science is insufficient for understanding them, since in life they function wholistically [Goldstein, 1995].

"To attempt to understand life from the point of view of the natural-science method alone is fruitless." ([Goldstein, 1995], 18)

> We have said that life confronts us in living organisms. But as soon as we attempt to grasp them scientifically, we must take them apart, and this taking apart nets us a multitude of isolated facts that offer no direct clue to that which we experience directly in the living organism. Yet we have no way of making the nature and behavior of an organism scientifically intelligible other than by construction out of facts obtained in this way. (27)

Goldstein argues that in order to understand complete organisms, one needs to balance fragmenting and symptomatizing methods from science with a more humanistic interest in how individuals function as a whole within the context of their lives. "Certainly, isolated data acquired by the dissecting method of natural science could not be neglected if we were to maintain a scientific basis. But we had to discover how to evaluate our observations in their significance for the total organism's functioning and thereby to understand the structure and existence of the individual person" (18). In practice, this means that Goldstein does not simply look for signs and symptoms, but tries to understand symptoms as fragmented manifestations of wholistic alterations to an individual nervous system that occur under disease. Statistics, he argues, is useless for this kind of understanding; instead, Goldstein works with case studies, "in which the historical, the personal, the experimental, and the clinical could all be brought together as a unity" ([Sacks, 1995], 8).

From Alienation to Wholism

The common thread in these solutions to the fragmentation of science is to combat alienation — the separation of scientist from object of science, the separation of different scientific subfields, the separation of the parts of the object being studied — by adding wholism to the toolbox of science. The object itself should be studied as a whole, its 'parts' being understood in relation to one another and to the object or organism as a whole. In addition, the object itself should be understood in context, a life world which includes the scientists studying the object. Rather than cutting the scientist off, the scientist and the object should be understood as in relationship to one another, leading to a 'personal' or 'subjective' science. In a wholistically informed science, 'objectivity' — in the sense of a natural world to be studied independently of the people who study it — is not possible; instead of objectivity, the goal of subjectivist science is, as Varela et. al. put it, *disciplined* knowledge [Varela *et al.*, 1991].

The alert reader may recognize the postulates of anti-boxology as expressed in Chapter 1. In essence, this chapter has been an articulation of the reasons for the anti-boxological approach. In the next chapter, we will look at the implications of this approach to science for AI, and in particular agent design. I will argue that autonomous agents, like schizophrenic patients, are cut off from their context; like assembly line workers, they are split into parts and rationalized until their overall actions lose any meaning. The result of these two moves is schizophrenia. To combat them, we can rethink AI's methodological strategies by importing the contextualizing approach of anti-psychiatrists and other critics of objectivist science. I call the resulting approach to AI "socially situated AI," and use it as the basis for rethinking agent design in the rest of the thesis. But first, we will take a short break to look at the system, the Industrial Graveyard, that both demonstrates the concepts of the analysis in this Chapter and provides the testbed for the technical work of the thesis.

Intermezzo I

The Industrial Graveyard

One of the heuristics we can derive from the previous chapter is that agents should be studied in the context in which they are used. In this Intermezzo, I will explain the system which provides the context for the agents developed for this thesis. This system, the *Industrial Graveyard*, is intended to demonstrate both the technical and the theoretical ideas of this thesis.

The Industrial Graveyard is a virtual environment in which a discarded lamp (the Patient, top right) ekes out a marginal existence in a junkyard. It is overseen by a nurse/guard (the Overseer, bottom right) from the Acme Sanitation and Healthcare Maintenance Organization. In this scenario, users are asked to take on the role of an auditor overseeing the efficiency of the Acme-run junkyard. Their job is to make sure the Overseer is sufficiently interceding in the Patient's existence. Here, I will describe the Industrial Graveyard both technically and in its connections to the theoretical ideas of the last chapter.

Introduction

The Industrial Graveyard is intended to make the user feel viscerally the constraints of objectivist knowledge production. There are two levels at which these constraints work. First of all, the Patient, for whom users generally develop a sense of pity, is shown caught within an industrial-institutional nightmare. The Patient has been discarded and lives in a fenced-in junkyard, in which its only companion is an Overseer who constantly punishes the Patient. The Patient is judged objectively by the Overseer, which is to say, without personal consideration of the meaning of the Patient's actions. When the Patient is no longer efficiently manageable, it is killed.

The second level at which the constraints of objectivist knowledge production are demonstrated is at the meta-level of the technology itself. In the rhetoric of virtual environments, you can do anything — be anyone — in a virtual world that lacks the limitations of everyday, physical existence. But technical systems always contain both consciously and unconsciously imposed constraints on what users can do, whether from the limitations of input technology (e.g., you can only take actions which correspond to a simple verb in the system's vocabulary) or simply because

ACME Sanitation & HMO

Welcome to our organization! We are proud to be the nation's largest and fastest growing Sanitation and Healthcare Maintenance Organization. We believe patients are best served through fast, efficient service. We strive to give patients what they need while minimizing costs through the reduction of extraneous services. You can be proud you have chosen to become part of a well-functioning health maintenance and sanitation machine.

Your input profile has indicated your appropriateness for supervisory position #45-TBKJ. This document contains instructions for your role.

You have been assigned to the Sanitation and Disposal sector. When patients can no longer function properly in their societal role, they can become a burden to themselves and those around them. Acme S&D is proud to take on the responsibility of their care. At the same time, in order to maintain profitability, dysfunctional patients must be monitored particularly closely, since they suffer from a chronic condition and as such may incur high costs over the lifetime of the patient. Patients are therefore assigned automated overseers which monitor their behavior. These overseers provide necessary care, but lack the human intuition to always determine when patient behavior is malignant. Your job is to provide back-up for the overseer, ordering it, when necessary, to monitor the patient more closely.

FIGURE I.1: User's introduction to the Industrial Graveyard

the authors did not think to program in some option that users can think of (have you ever tried to make friends with the monsters in Doom?). Because virtual environments are often presented in current rhetoric as authorless — as real worlds, not personal visions — they, too, are a form of objectivist knowledge production.

In this sense, the Industrial Graveyard can be understood as a *parody* of a virtual environment. The function of parody in the system is to make objectivist construction of technical artifacts, which is normally a theoretical construct, be experienced in a visceral sense by users, becoming part of their subjective experience. This is done by exaggerating the constraints of the system to the point where users are forced to become aware of them. Far from being able to be anyone or do anything, users are told exactly what they are expected to do, and the system is designed to try to make them uncomfortable in the role to which they are assigned. The 'cartoony' nature of the world, in contrast with the photographic physical realism of many virtual environments, is also intended to communicate that the world was written by someone.

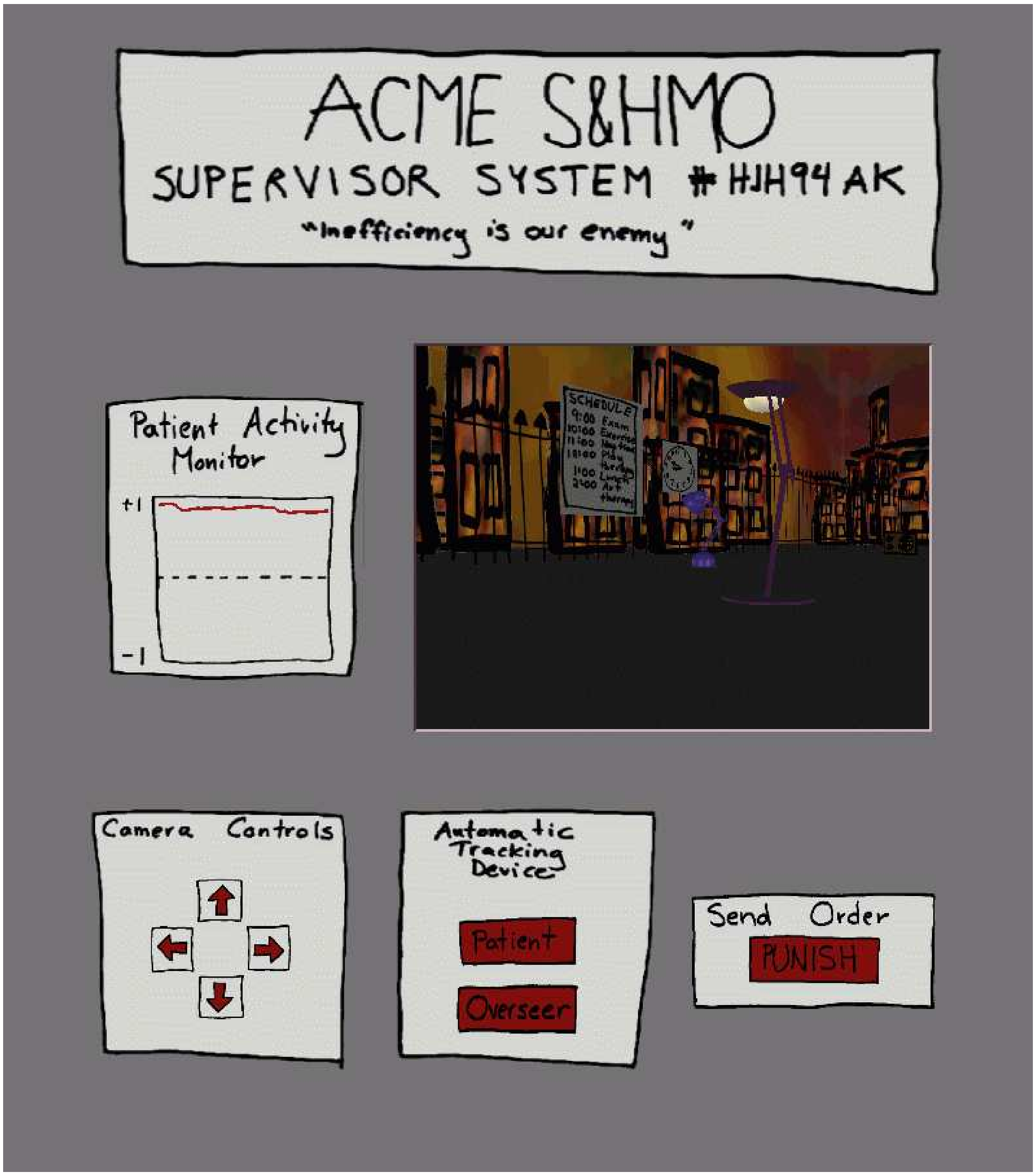


FIGURE I.2: The presentation of the system.

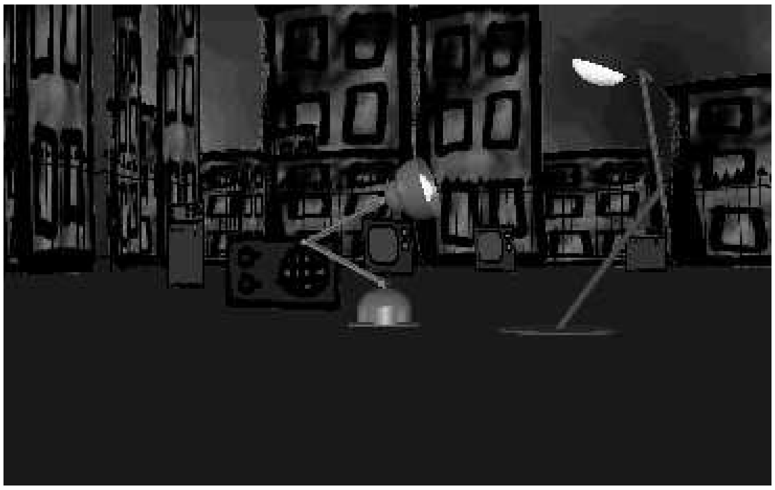


FIGURE I.3: The Overseer prepares to attack the Patient.

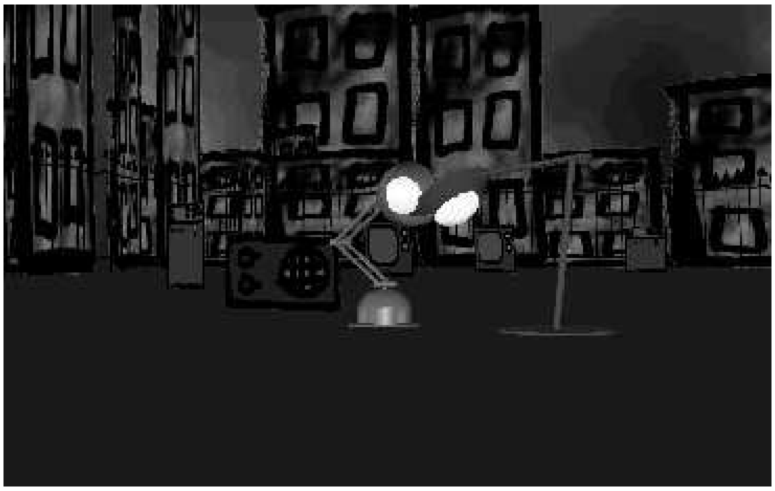


FIGURE I.4: The Overseer strikes the Patient.

Design

Users are introduced to the system through a set of written instructions that explain their role (Figure I.1). The instructions repeat the themes of industrialized, mechanized culture, and place users in a position of colluding with the Overseer against the patient.

Users then 'interact' with the system, which is illustrated in Figure I.2. The word 'interact' is in quotation marks because users' ability to influence the system is minimal. They can move the "surveillance camera" around (although it stays within a fenced-off area), and they can order the Overseer to harass the Patient. There is nothing users can do to help the Patient.

To the left of the view into the junkyard is a graph which shows the user how good or bad the Patient is being. 'Goodness' is calculated objectively by measuring the amount of movement of the angles of the

FIGURE I.5: The aftermath of the attack.

Patient's body. The Patient is considered optimally 'good' when it is frozen in place. When the Patient becomes 'bad' — by, for example, being excited about exploring the junkyard — the Overseer comes over and strikes the Patient (Figures I.3-I.4), turning the Patient off (Figure I.5).

Plot

The Industrial Graveyard includes a kind of story — the Patient is deposited in the junkyard, explores it (under constant interruptions by the Overseer), and, eventually, is killed by the Overseer. The 'plot' of the Industrial Graveyard follows Tinsley Galyean's notion of interactive narrative flow: it accommodates users' actions and random variations in the agents' behavior without fundamentally altering the story [Galyean, 1995]. Variations occur in the timing of the plot points and in how they are realized, but, no matter what, the same basic plot points always occur.

The story is maintained using the concept of "story stages," which are component pieces of the story in a sequential order. The current stage is stored in a data structure which is accessible to both characters. When a character does something to advance the story to the next stage, the character updates the data structure to reflect the new story stage. Both characters modify their behavior according to the stage the story is in.

To start out with, the Patient is dropped into the world, landing in a diagnostic machine. The Overseer comes over and reads the Patient's diagnosis, while the Patient cowers (Figure I.6). After the Overseer leaves, the Patient looks around, and gingerly steps out into the junkyard.

The Patient wanders around the junkyard, looking at the objects in it, and trying to stay away from the Overseer, who regularly harasses it. Eventually, the Patient notices a schedule of activities posted on the fence. It becomes engrossed in reading it (Figure I.7), oblivious to the Overseer, who comes up behind it. The clock turns 10 (time to exercise, according to the schedule), and the Patient, noticing the Overseer, frantically starts exercising.

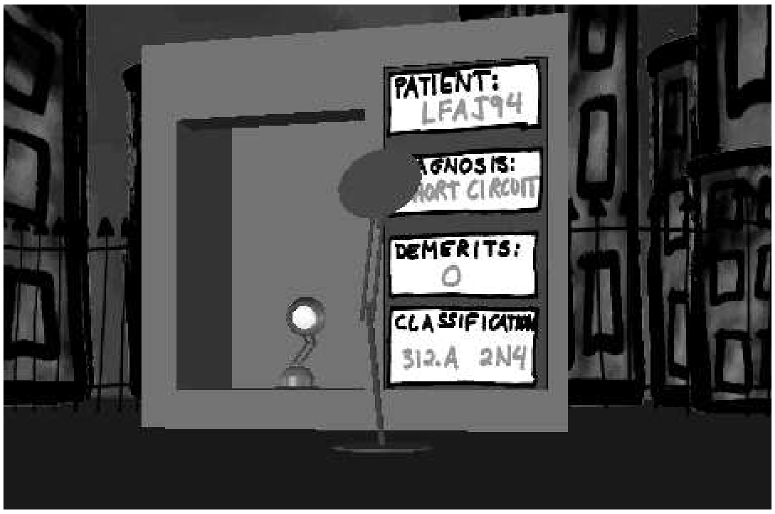


FIGURE I.6: The Patient is examined in the monitor.

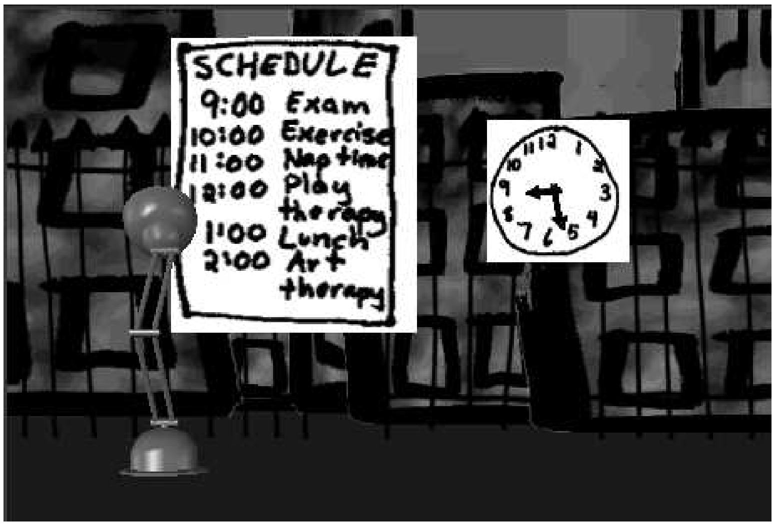


FIGURE I.7: The Patient reads the schedule.

After the Overseer leaves, the Patient loses interest in exercise and wanders off sadly. It stands by the fence, sighing and looking out at the world that has rejected it. Suddenly, the Patient's light goes out. The Patient shakes its head, trying to get the light on. When that doesn't work, the Patient starts hitting its head on the ground, trying to fix the short circuit (Figure I.8). It gets more and more frantic, banging around more and more — and therefore, by the logic of the Overseer, being more and more bad.

Finally, the Overseer comes over. The Patient cowers, wondering what is going on. The Overseer brings a large mechanism over the Patient's head, from which a beam emerges (Figure I.9). When the beam recedes, only the Patient's corpse is left (Figure I.10).

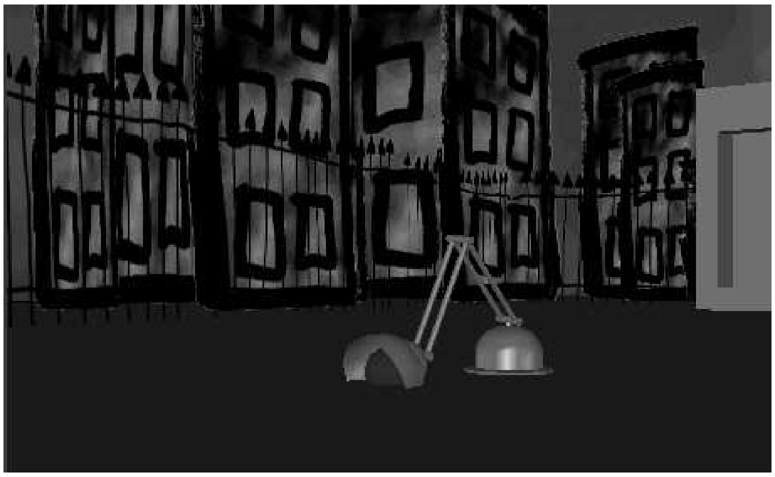


FIGURE I.8: The Patient hits its head on the ground.

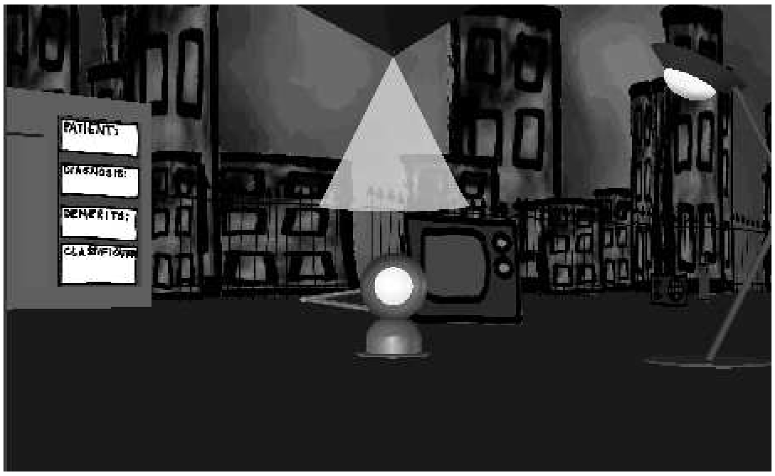


FIGURE I.9: The Patient being struck by the beam.

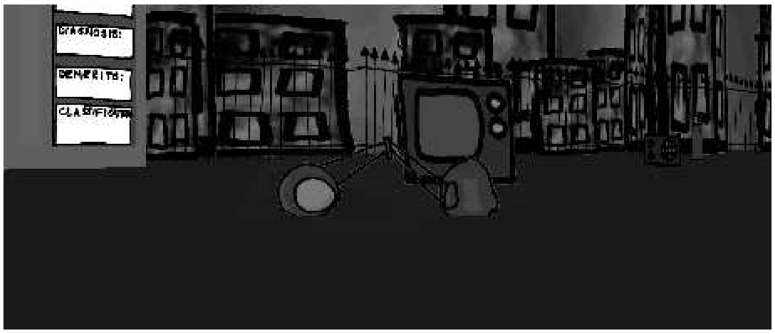


FIGURE I.10: The happy ending.

Construction

The Industrial Graveyard is built on the skeleton of a previous Oz group system, the *Edge of Intention* [Loyall and Bates, 1993]. In order to create the Industrial Graveyard, I removed the Edge of Intention's hand-coded graphics system (bouncing balls in a 2.5-D environment with fixed camera position) and replaced them with an interface to the Inventor 3D graphics toolkit [Wernecke, 1994]. The agent's bodies and world are Inventor models which can be loaded and reconfigured at run-time; the user's viewing position is a movable camera immersed in the world, rather than a God-like view from above. The 'cartoony' flat objects in the Industrial Graveyard are created by projecting transparent texture maps onto flat planes.

The Overseer's behavior is written in Bryan Loyall's Hap [Loyall and Bates, 1991], while the Patient is written in the *Expressivator*, the system I will describe in Chapters 5 and 7. Each agent architecture sends low-level commands ("spin," "jump," "move eyes") to a motor system which models the creature's bodies. An underlying physical simulation implements actions by modelling the lamps as Edge-of-Intention-style bouncing spheres. The system runs in real time on an SGI Indigo 2. Most of the running time is devoted to graphics; the Patient's mind takes about 14 milliseconds per frame, while the graphics takes about 77. Most of the graphics time is devoted to texture mapping.

Chapter 4

Socially Situated AI

The cultural theory analysis of Chapter 3 suggests that industrialization and institutionalization share properties that lead to schizophrenia. Both industrialization and institutionalization take objective views of living beings. By 'objective,' I mean that they are taken out of their sociocultural context and reduced to a set of data.[1] Because these data are not related to one another or the context from which they sprung, the result is a fragmentation of experience that cultural theorists term schizophrenia.

Cultural theory therefore suggests that, in order to address schizophrenia, we can take the *opposite* approach. Rather than seeing workers or patients as objects to be manipulated or diagnosed, we could see them *subjectively*. This means turning objectivity as defined above on its head: studying people in their life context and relating the things we notice about them to their existence as a whole.

If you are a technical researcher, it is quite possible that Chapter 3 was an insurmountable struggle, or at the very least left you with lingering doubts about the accuracy or validity of the cultural theory argument. But however you feel about the understandability or truth-value of that argument, the perspective cultural theory brings can be understood as a kind of heuristic which could be tried out in AI. At this level, cultural theory suggests the following: *if your agents are schizophrenic, perhaps you need to put them in their sociocultural context.*

In this chapter, we'll explore what it means for an agent to be designed and built with respect to a sociocultural environment. This way of doing AI I term *socially situated AI*. I will differentiate socially situated AI from the approaches taken in classical and alternative AI, and then discuss the impact this methodological framework has on the way AI problems are defined and understood. This different way of doing AI will become the key to solving schizophrenia in Chapters 5 and 7 by suggesting the redefinition of the problem of schizophrenia as a difficulty of *agent communication* rather than of *internal agent structure* — thereby finding a trapdoor to get us out of the Catch-22 of schizophrenia and atomization.

[1] The notion of what exactly objectivity means in various fields and usages is a quagmire in which, at the moment, I prefer not to be morassed. Please accept this usage of objectivity as a definitional statement of what I mean by 'objectivity' here, as opposed to a pronouncement of what anyone would mean by it.

AI in Context

The heuristic suggested by cultural theory — that agents should be considered with respect to their context — should have a familiar ring to technical researchers. The contextualization of agents, i.e. their definition and design with respect to their environment is, after all, one of the major bones alternativists like to pick with classicists. Alternative AI argues that agents can or should only be understood with respect to the environment in which they operate. The complexity or 'intelligence' of behavior is said to be a function of an agent *within* a particular environment, not the agent understood in isolation as a brain-in-a-box.[2]

But the contextualization which is so promoted in alternative AI is actually limited, in particular by the following implicit caveat to its methodology: *the agent is generally understood purely in terms of its* physical *environment* — *not* in terms of the sociocultural environment in which it is embedded. Generally speaking, alternativists examine the dynamics of the agent's activity with respect to the objects with which the agent interacts, the forces placed upon it, and the opportunities its physical locale affords. Some alternativists have also done interesting work examining the dynamics of agent activity in *social* environments, where 'social' is defined as interaction with other agents. They generally do not, however, consider the *sociocultural* aspects of that environment: the unconscious background of metaphors upon which researchers draw in order to try to understand agents, the social structures of funding and prestige that encourage particular avenues of agent construction, the cultural expectations that users — as well as scientific peers — maintain about intentional beings and that influence the way in which the agent comes to be used and judged.

In fact, when such aspects of the agent's environment are considered at all, many alternativists abandon their previous championing of contextualization. They see these not-so-quantifiable aspects of agent existence not as part-and-parcel of what it means to be an agent in the world, but as mere sources of noise or confusion that obscure the actual agent. They may say things like this: "The term 'agent' is, of course, a favourite of the folk psychological ontology. It consequently carries with it notions of intentionality and purposefulness that we wish to avoid. Here we use the term divested of such associated baggage" ([Smithers, 1992], 33) — as though the social and cultural environment of the agent, unlike its physical environment, is simply so much baggage to be discarded.

In this respect, the alternativist view of agents-in-context is not so different from the Taylorist view of worker-in-context or the institutional view of patient-in-context. After all, Taylorists certainly look at human workers in context; in the terminology of situated action, they analyze and optimize the ongoing dynamics of worker-and-equipment within the situation of a concrete task, rather than the action of the worker alone and in general. Similarly, institutional psychiatrists look at human patients in context; they are happy to observe and analyze the dynamics of patient interaction with other people and objects in the world, as long as in those observations and analyses they do not need to include themselves. In each of these cases, contextualization is stopping at the same point: where the

[2] Classicists will recognize the same argument as Simon's ant.

social dynamics between the expert and the object of expertise, as well as its *cultural* foundation, would be examined.

I do not believe that the elision of sociocultural aspects from the environment as understood by alternative AI is due to any nefarious attempt to hide social relations, to push cultural issues under the rug, to intentionally mislead the public about the nature of agents, etc. Rather, I believe that because AI is part of the scientific and engineering traditions, most alternativists simply do not have the training to include these aspects in their work. In Chapter 3, I noted that science values simplification through separation, and one of the key ways in which this is done is by separating the object of study from the complex and rich life background in which it exists. This strategy lets researchers focus on and hopefully solve the technical problems involved without getting bogged down in all kinds of interconnected and complex issues which may not have direct bearing on the task at hand.

The Return of the Repressed

The problem, though, is that even from a straightforward technical point of view, excluding the sociocultural context is sometimes unhelpful. At its most basic, ignoring this context does not make it go away. What ends up happening is that, by insisting that cultural influences are not at work, those influences often come back through the back door in ways that are harder to understand and utilize.

As an example, consider the use of programming through the use of symbols. Symbolic programming involves the use of tokens, often with names like "reason," "belief," or "feeling" which are loaded with cultural meaning to the agent designer. Critics point out that the meaningfulness of these terms to humans can obscure the vacuousness of their actual use in the program. So a programmer who writes a piece of code that manipulates tokens called 'thoughts' may unintentionally lead him- or herself into believing that this program must be thinking.

Alternative AI, generally speaking, involves a rejection of these sorts of symbols as tokens in programs. This rejection is often based on a recognition that symbolic programming of the kind classical AI engages in is grounded in culture, and that symbols carry a load of cultural baggage that affects the way programs are understood. Some of them believe that by abandoning symbolic programming they, unlike classicists, have also abandoned the problem of cultural presuppositions creeping into their work. And, in fact, it is true that many alternative AI programs do use such symbols sparingly, if at all, in their internal representations.

Nevertheless, it would be fair to say that the architecture of such agents involves symbols *to the extent that the engineer of the agent must think of the world and agent in a symbolic way in order to build the creature*. For example, the creature may have more or less continuous sensors of the world, but each of those sensors may be interpreted in a way that yields, once again, symbols — even when those symbols are not represented explicitly as a written token in an agent's program. For example, a visual image may be processed to output one of two control signals, one of which triggers a walking style appropriate when on carpets, and one of which triggers a walking style appropriate when

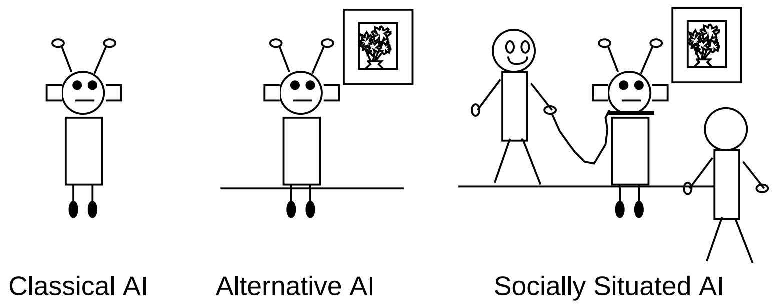


FIGURE 4.1: The increased context from classical through alternative to socially situated AI.

not on carpets. While a variable named 'on-carpet' may not appear in the agent's code, it would be fair to predicate an 'on-carpet' symbol *in the designer's thinking* as s/he constructed the agent - a symbol which is as informed by the designer's cultural background as the identifiable 'on-carpet' symbol in a classical AI program.

The behaviors into which the agent is split up are similarly fundamentally symbolic ("play fetch," "sleep," "beg," etc.) and are influenced by cultural notions of what behaviors can plausibly be. While alternative AI has gotten away from symbolic representations within the agent when seen in isolation, it has not gotten away from symbolic representations when the agent is seen in its full context. Once you look at the entire environment of the agent, including its creator, it is clear that despite the rhetoric that surrounds alternative AI, these symbols — and their accompanying sociocultural baggage — still play a large role.

Leaving out the social context, then, is both epistemologically inadequate and obfuscating. By not looking at the subjective aspects of agent design, the very nature of alternative AI programming, as well as the origin of various technical problems, becomes obscured. This is particularly problematic because not being able to see what causes technical problems may make them hard, if not impossible, to solve. We will see in the next chapter that this is exactly what happens with schizophrenia — and that by taking the opposite tack a path to solution becomes much more straightforward.

Socially Situated AI

What should AI do instead? Alternativists believe that situating agents in their physical context often provides insight into otherwise obscure technical problems. I propose that we build on this line of thinking by taking seriously the idea that the social and cultural environment of the agent can also be, not just a distracting factor in the design and analysis of agents, but a valuable resource for it (Figure 4.1. I coined the term 'socially situated AI' for this method of agent research.

Here, I will first describe at a philosophical level the postulates of socially situated AI. This lays out the broad framework within which technical work can proceed. I'll then discuss at a more concrete level what it means to design and build agents with respect to their sociocultural context. This concrete description will form the basis for redefinition of schizophrenia in the next chapter.

Postulates of Socially Situated AI

Like other methodological frameworks, including classical and alternative AI, socially situated AI involves, not just a kind of technology, but a way of understanding how to define problems and likely avenues of success. I represent this changed way of thinking here through an enumeration of postulates of socially situated AI. These are propositions that form the framework for how research is done and evaluated. Specifically, socially situated AI distinguishes itself from other forms of AI through explicit commitment to the following principles:

1. *An agent can only be evaluated with respect to its environment, which includes not only the objects with which it interacts, but also the creators and observers of the agent.* Autonomous agents are not 'intelligent' in and of themselves, but rather with reference to a particular system of constitution and evaluation, which includes the explicit and implicit goals of the project creating it, the group dynamics of that project, and the sources of funding which both facilitate and circumscribe the directions in which the project can be taken. An agent's construction is not limited to the lines of code that form its program but involves a whole social network, which must be analyzed in order to get a complete picture of what that agent is, without which agents cannot be meaningfully judged.

2. *An agent's design should focus, not on the agent itself, but on the dynamics of that agent with respect to its physical and social environments.* In classical AI, an agent is designed alone; in alternative AI, it is designed for a physical environment; in socially situated AI, an agent is designed for a physical, cultural, and social environment, which includes the designer of its architecture, the creator of the agent, and the audience that interacts with and judges the agent, including both the people who engage it and the intellectual peers who judge its epistemological status. The goals of all these people must be explicitly taken into account in deciding what kind of agent to build and how to build it.

3. *An agent is a representation.* Artificial agents are a mirror of their creators' understanding of what it means to be at once mechanical and human, intelligent, alive, what cultural theorists call a subject. Rather than being a pristine testing-ground for theories of mind, agents come overcoded with cultural values, a rich crossroads where culture and technology intersect and reveal their co-articulation. This means in a fundamental sense that, in our agents, we are not *creating* life but *representing* it, in ways that make sense to us, given our specific cultural backgrounds.

Socially Situated AI as Technical Methodology

These philosophical principles do not necessarily give technical researchers much to go on in their day-to-day work. Concretely speaking, socially

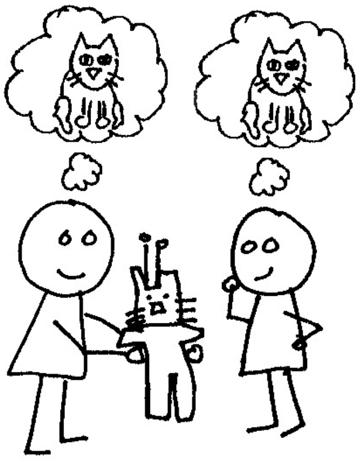


FIGURE 4.2: Agents as communication.

situated AI can be understood in the following way. Rather than seeing an agent as a being in a social vacuum, we can see it as represented in Figure 4.2: as a kind of *communication* between a human *designer* who is using it to embody a conception of an agent and a human *audience* who is trying to understand it.

After all, for many applications it is not enough for an agent to function correctly in a technical sense. Many times, the agent should also be *understandable*. For example, when an agent researcher designs an artificial cat, s/he will have some ideas about the kinds of behaviors the cat should have and the kind of motivations behind the cat's selection of various behaviors — ideas which, optimally and sometimes crucially, the viewers of the agent should also pick up on. In this sense the agent as program is a kind of vehicle for a conception of a particular agent, which is communicated from the agent-builder through the technical artifact to the observers of or interactors with the agent.

This way of understanding socially situated AI can be thought of as a change in metaphor. Many current approaches to AI are based on the metaphor of *agent-as-autonomous*: the fundamental property of such an agent is its basic independence from its creator or users. Lenny Foner, for example, defines autonomy as one of the most basic aspects of being an agent.

> Any agent should have a measure of autonomy from its user. Otherwise, it's just a glorified front-end, irrevocably fixed, lock-step, to the actions of its user. A more autonomous agent can pursue agenda independently of its user. This requires aspects of periodic action, spontaneous execution, and initiative, in that the agent must be able to take preemptive or independent actions that will eventually benefit the user. [Foner, 1993]

This autonomy implies that the agent's fundamental being is as a thing-for-itself, rather than what it actually is: a human construction, usually a tool. AI researchers are far from believing that agents magically spring from nowhere, and autonomy can certainly be a useful notion. Nevertheless, the focus on autonomy — separation from designer and user — as a *defining* factor for agents can unwittingly hide the degree to which both designers and users are involved in the agent's construction and use.

As an alternative to this metaphor, socially situated AI suggests the metaphor of *agent-as-communication*. Socially situated AI sees agents not as beings in a vacuum, but as representations which are to be communicated from an agent-builder to an audience. This point of view is deeply informed by recent work in believable agents such as [Neal Reilly, 1996] [Loyall, 1997a] [Wavish and Graham, 1996] [Blumberg and Galyean, 1995], which focus more and more on the audience's perception of agents, rather than on an agent's correctness per se. This conception of agents is also very like contemporary conventional conceptions of artwork, as vehicles through which ideas can be transmitted from a designer to his or her audience.

But the concept of agent-as-communication is not limited to believability or other 'artsy' applications. This is because proper perception of agents matters not only when we want to communicate a particular personality through our agents. It matters in *any* situation where the design of the agent — including its purpose, methods, functions, or limitations — should be understood by the people with which the agent interacts.

Thinking of agents as communication has several advantages. By making the commitment that 'agentiness' is meant to be communicated, we can explicitly communicate to the audience what the agent is about, rather than assuming (often incorrectly) that this will happen as a side-effect of the agent "doing the right thing." And by building agents with an eye to their reception, builders can tailor their agents to maximize their effectiveness for their target audience. In this sense, agents built for social contexts can be not only more engaging but more *correct* than purely rational, problem-solving agents, in the following sense: they may actually get across the message for which they have been designed.

This change in metaphor from autonomy to communication will have crucial implications, both in redefining the problem of schizophrenia in the next chapter, and for agent architecture down to its very details, as we will see later. It will turn out that behavior-based technology is so heavily invested in the metaphor of agent-as-autonomous that switching to agent-as-communication will have ramifications throughout the agent architecture. In the next chapter, we will see that taking seriously the quality of agent communication means redefining even the basic building blocks of behaviors as *signifiers*. In Chapter 7, we will learn that communication of agent motivation necessitates the use of *transitions* to explain to the user the agent's normally implicit decision-making process. But before we get to these changes, we will go back to the technical problem of schizophrenia with which we started, and look at how socially situated AI redefines the relationship between schizophrenia and atomization, showing us a way out of the conundrums of Chapter 2.

Chapter 5

Architectural Mechanisms I: Transitions as De-Atomization

Socially situated AI as a theory is well and good, but the proof of the pudding is in whether it actually helps us do anything differently. This chapter is devoted to exploring the technical consequences of the theoretical framework we have been developing in the last 2 chapters. Conceptually, we will start by rethinking the technical problem of schizophrenia as defined in Chapter 2 from the vantage point gained by the forays we have made into humanism. This new conception immediately suggests that the problem of schizophrenia should be rephrased. Instead of looking at schizophrenia as a property of agent code, we will look at schizophrenia as a problem of agent communication.

This way of rephrasing of the technical problem is amenable to more-or-less straightforward technical solution. I will use conventional AI techniques to solve this problem, leading to the following architectural innovations:

1. Behaviors are re-understood as *signifiers*, which explicitly act to communicate the agent's activity to users through the use of low-level *signs*. A *sign-management system* allows the agent to keep track of which signs and signifiers have been communicated to the user, so that the agent can make behavioral decisions based not only on what *it* thinks it is doing, but also on the likely user impression of its activities so far.

2. Sudden breaks between these signifying behaviors are smoothed over using *transitions*. Instead of leaping from behavior to behavior in the manner of the schizophrenic agents of Chapter 2, the agent gradually morphs between them.

3. These transitions are implemented using *meta-level controls*, which allow behaviors to share information and coordinate their effects. By making the coordination of behaviors explicit — rather than an implicit side-effect of the underlying architecture — meta-level

controls allow the relationships between behaviors to be expressly communicated to the user.

If you are not technically trained this chapter can be viewed as a case study in AI methodology. Since the rhetorical style of AI argumentation is not always transparent to those not trained in AI, margin boxes will provide some context by explaining the role of each piece in developing the larger argument. AI researchers may also find this outsider perspective on AI argumentation enlightening!

It will turn out that this basically purely technical approach works to smooth observable behavior together, thereby making agents seem less schizophrenic. Unfortunately, that in itself will not necessarily help make agents that are effective in appearing truly intentional. To put it simply, the techniques developed here may keep the agents from looking transparently bad (which is of course nice), but they don't necessarily make them look particularly good. For that, we will need to think more deeply about the assumptions and requirements of the technical approach. We will do this through another foray into animation (Intermezzo II) and psychology and the cultural studies of science (Chapter 6). These will allow us to build on the technical foundations of this chapter to create the full agent architecture in Chapter 7.

Socially Situated AI vs. Good Old-Fashioned Alternative AI

The technical developments in this chapter depend in a deep sense on understanding how socially situated AI fundamentally changes the ground on which alternative AI operates. As discussed in the previous chapter, socially situated AI suggests that the agent and its behavior should be thought about, not in terms of the agent itself, but in terms of communication between the designer of the agent and its audience. Rather than *intelligent* agents, then, the focus is on creating *intelligible* agents, ones that successfully communicate to the audience the idea for the agent that the designer had in mind.

This switch from intelligence to intelligibility may be recognizable to AI researchers as the mindset change behind believable agents that motivates such work as [Bates, 1994], [Loyall, 1997a], and [Neal Reilly, 1996]. Believable Agents — characters that are intended to communicate a particular artist-chosen personality — similarly focus on situated communication over an agent's abstract (and perhaps uncommunicated) reasoning abilities. Socially situated AI builds on a rich foundation laid by Believable Agent researchers, by seeing this communication perspective as not only useful for agents that are to inhabit works of art or entertainment, but for *all* agents — whether intended as living creatures or as helpful tools — whose activity should be comprehensible to humans with which it interacts. This may include agents like office robots, tele-autonomous systems, or automated flight systems, whose *function* is totally utilitarian, but whose actions should be understandable in order to function well with and to inspire confidence from the humans who come into contact with them.

Believable Agents researchers have long pointed out that the nature and utility of various technological mechanisms may change radically when the intelligibility of agents is seen as equally important to — or more important than — their reasoning abilities in abstract. Taking this point of view changes, for instance, what behaviors fundamentally mean. In alternative AI, behaviors are assemblages of actions that help the agent to fulfill its goals with respect to the environment, e.g. to navigate around the room (Brooks), avoid getting too hungry (Blumberg), or to kill enemies and win the video game (Agre and Chapman). Behaviors are defined in terms of their correctness in helping agents to achieve their goals.

In socially situated AI, however, behaviors are fundamentally the designer's vehicle for communicating an idea of agent activity to the audience. Behaviors need to be designed, not just in terms of fulfilling the internal goals of the agent, but in terms of what the agent is communicating to the audience. It is not enough to just do something; the audience must be able to tell the agent is doing it. This means a behavior includes the intention to communicate that behavior to the audience. 'Behaviors' therefore explicitly become something more like 'understandable aggregates of action' than the a priori, problem-solving modes of behavior in most behavior-based AI applications.

Many behavior-based researchers have focused on action selection, i.e. determining when an agent should switch to another behavior. Again, action-selection takes a problem-solving view of agents in that it focuses on correctness: when the agent should, for the sake of correctness, switch to a different behavior. The focus on agent *presentation* that is part and parcel of socially situated AI means that the question of what behavior the agent should pick is less important than how well the agent communicates through that behavior. For socially situated AI, then, the fundamental problem is better rephrased as what Tom Porter terms *action-expression* [Porter, 1997] [Sengers, 1998]:

> How can the agent at every point choose an action that best communicates the goals, activities, and emotions the designer has selected to its audience?

But even this point of view is too limiting, since it causes us to focus on the mechanics of agent action choice. The point here is not doing the "correct" behavior, but doing the behavior well. For human understanding, the manner in which the agent *does* the what it has chosen is just as important, if not more so, then whether or not the agent has chosen the optimal thing to do. These conceptual differences are summarized in Figure 5.1. These differences form the foundation for addressing schizophrenia.

Schizophrenia Revisited

In Chapter 2, we learned that schizophrenia comes about when the agent's behaviors are so atomized that they become easy for the user to pick out. Schizophrenic behavior has one or more of the following properties:

	Alternative AI	Socially Situated AI
Concept of agent	Autonomous	Communication
Concept of behaviors	Chunks of problem-solving	Chunks of meaning
Fundamental problem	Action-selection	Action-expression (and more)

FIGURE 5.1: Differences between alternative and socially situated AI

1. The agent, rather than engaging in a fluid stream of activity, jumps abruptly from behavior to behavior.
2. The agent combines actions from different behaviors in a way that makes no overarching sense.

These properties happen because the agent's behavior is atomized into meaningful units, with very little intercoordination of each unit.

A natural instinct when faced with schizophrenia is to hope it can be resolved by getting rid of atomization. It turns out that this is probably not a very practical solution for complex agents with a variety of high-level behaviors. Atomization, in the form of modularization, is what allows us to build these complex systems in the first place, since unmodularized systems beyond a certain size become an interrelated, undebuggable mess.[1] There are natural limitations to the size of these unmodularized systems because people simply cannot keep track of what is going on in the code without some level of abstraction.

In Chapter 2, we came to the conclusion that schizophrenia is therefore unsolvable. This is, in fact, the case, as long as we look at the agent in isolation. A humanly constructed agent will almost certainly be atomized, and therefore also schizophrenic. However, the problem of schizophrenia changes in some interesting ways when looked at in the context of agent and designer.

Situating Schizophrenia in Context

Here, I describe the fundamental philosophy motivating the technical choices I make later. You might have thought that fundamental philosophy is in Chapter 4, and how right you are! Here, the goal is to bring that philosophy close to the technology so that it can be instantiated.

From the designer's point of view, atomization is necessary in order to maintain a manageable system. Constructed agents don't spring out of the air; they are constructed by someone who needs to be able to understand and control how they work. In order to be effective, the agent architecture must be simple enough that the designer can understand and, to a reasonable extent, control the effect of the agent. This leads to the first heuristic we will use in addressing schizophrenia:

Remember the designer
Support modularized code to make the programming job easier and more understandable.

[1] It is possible that such systems could be learned automatically. The exploration of mechanisms which could automatically generate complex, expressive, and deeply interrelated behavior is still in its infancy. I suspect (but certainly cannot prove) that systems that are truly so complex will also have to be learned step-by-step in a modularized fashion that may undermine the possibility for truly interrelated, learned behavior. The argument in this thesis limits itself to systems which are (mostly) humanly designed and built.

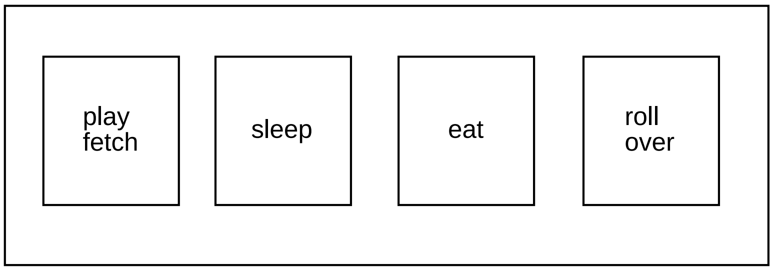


FIGURE 5.2: Atomized behaviors leave gaps that are obvious to the user.

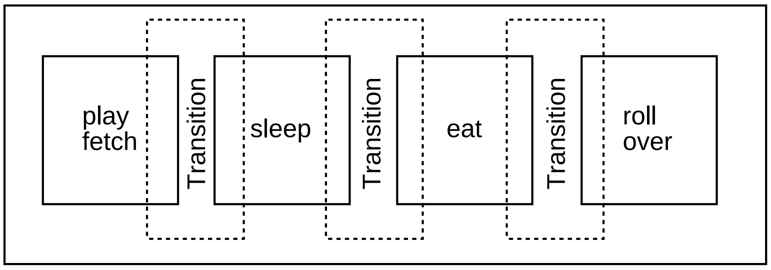


FIGURE 5.3: Atomized transitions cover up the breaks left by atomized behaviors.

As noted before, while atomization is good for the designer, it is bad for the user; the agent jumps abruptly from behavior to behavior, or mixes its actions together in an incoherent mess. If we do not want to give up atomization, we need to find a way to mitigate its effect. Specifically, since the agent is a form of communication, our goal will be to integrate the *effect* of the agent, rather than the agent per se. This forms the basis of our second heuristic:

Remember the audience Integrate the *impact* of the behavior, not behaviors themselves.

This observation holds the key to solving schizophrenia. From the user's point of view, atomization is bad because it makes it too easy to see the 'breaks' in the system. The problem for the user is that he or she can see the 'lines' the programmer has drawn in the agent. Those lines are obvious, since they are drawn between the behaviors, i.e. the high-level activities we expect and hope the user will be able to recognize. These considerations lead to an obvious conclusion: *if we draw the lines somewhere different* — somewhere where the user is not trained to look, and hence has more difficulty recognizing them — *the agent may not appear as schizophrenic*.

In particular, if users are good at recognizing behaviors, abrupt switching between behaviors will be obvious to them. Instead, we should switch *during* a behavior. When the switches occur during a behavior, not between behaviors, they will be less obvious to people watching the agent, since even after the switch the agent is, from the point of view of the audience, still doing the same thing. The way to do this is to make behavior transitions — which traditionally fall through the architectural cracks — into full-fledged modules or components of the agents, i.e. *atomize the*

Of course, these figures don't prove anything. They rely on a visual metaphor to make the basic argument plausible. Such diagrams have a venerable tradition in AI... as well, it seems, as everywhere else.

behavior transitions themselves. This concept is graphically represented in Figures 5.2 and 5.3.

Since this is the key to all the technical work in this chapter, I will leave the reader a moment of silence to contemplate this changed viewpoint.

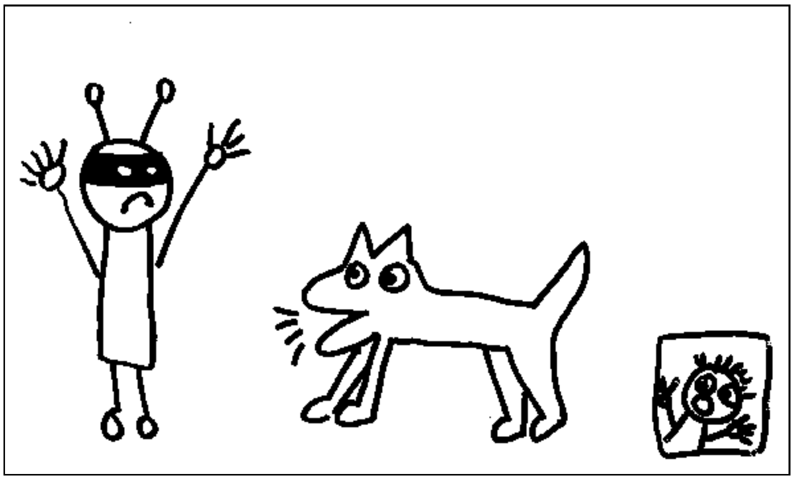


FIGURE 5.4: Our dog in action

Treating Schizophrenia in Attack Dogs

For example, suppose we want to build an artificial "guard dog" Following the behavior-based approach, we'll pick a selection of behaviors for it, such as "eat," "sleep," "chew on bone," and, since it is a guard dog, "bark at intruder." Then, we'll try to find the circumstances under which each behavior is appropriate: if you're hungry, eat; if you're tired, sleep; if there is a bone, chew on it; if there is an intruder, bark.

Now imagine that one day our dog has found a burglar to bark at (Figure 5.4 — the user is represented in the box in the corner). In this case, having been properly programmed, the dog starts barking. The observer, having some background knowledge of dogs and burglars, is likely to understand that the dog is trying to scare away the intruder.

Suddenly, the dog realizes that it has gotten very tired (Figure 5.5). What this means in technical terms is that the internal counter for "tired" has reached a threshold that outweighs the importance of scaring away the burglar (maybe the dog has been barking at various intruders all day, or had a particularly thrilling morning at the park).

Since sleeping is now the most important thing to do, the dog immediately stops barking and passes out on the floor (Figure 5.6). This sudden change of circumstances leaves the poor observer stymied: what on earth is that dumb dog doing? is it dead? did the burglar drug it? does the burglar have mystical hypnotic powers? This strange sudden break is, for the observer, the symptom of the dog's schizophrenia.

By adding a transition, we can mitigate the effects of this schizophrenia on the audience. A transition could work like this. As soon as the dog starts noticing that it is getting tired and likely to switch to sleeping, the dog will terminate the bark-at-intruder behavior and start a bark-to-sleep transition. This transition would keep the dog barking, while gradually adding some signs of sleepiness. When the dog becomes very tired, the dog could become more droopy, bark more slowly, lie down, bark a few more times, yawn, and then fall asleep. With this transition, there is no sudden break to confuse the user; the user understands both what the agent is doing (sleeping, not dead), and why the agent did it (was very

In this section, I will use a specific example to make the technical proposal plausible. The idea is to spin a narrative under which the technology seems intuitively correct. It connects a particular technology with a suggested way of experiencing reality. Since AI researchers are also human beings, I like this way of connecting lived experience with technology, in the philosophy of [Varela *et al.*, 1991].
Nevertheless, this strategy is also subject to some abuse (a nice analysis of this phenomenon can be found in [Agre, 1990]). In the 70's, a relatively common technique was to have one or two examples to intuitively ground the technology, and never bother to implement it at all (in all fairness, one can hardly blame people for trying to avoid working on the complicated and slow machinery of that day). Another tactic was to implement only a single example of the basic idea, and proclaim that as some kind of proof. This approach is pleasantly satirized in [McDermott, 1981].
The '90's have, for various reasons including this, seen a kind of backlash against this style of AI. Now, researchers often insist on concrete justification, preferably of an objective, empirical kind inspired by physics. Hopefully, we will one day be able to find a happy medium.

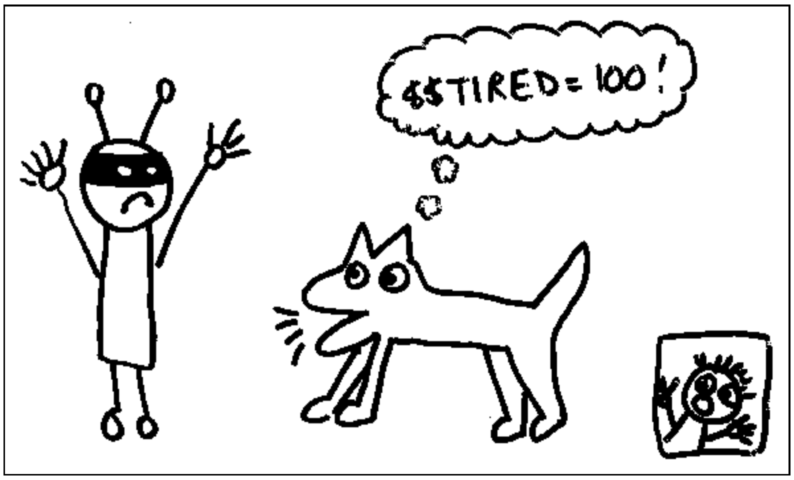


FIGURE 5.5: Rex gets sleepy

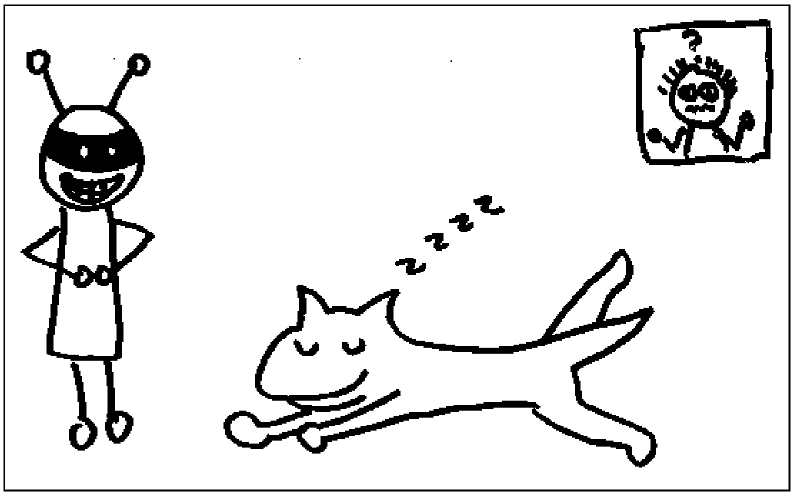


FIGURE 5.6: Rex immediately starts to snooze

tired). The transition "covers up" the break between the behaviors, so the change in the agent's behavior is gradual and natural. It is likely the user will not notice the "real" (i.e. internal) breaks at all (the one between bark and the transition, and the one between the transition and sleeping).

Summary

Behavior transitions can be thought of as a a form of strategic de-atomization. Rather than getting rid of atomization at the code level (where the designer needs it), behavior transitions reduce the apparent atomization of the agent from the audience's point of view. Behavior transitions allow the designer to use the full strength of atomized high-level behaviors without reducing agent activity to an abrupt jumping around from behavior to behavior. Agents with behavior transitions do not have discrete behavior breaks; rather, they *blend* their behaviors together.

Technically speaking, behavior transitions are a straightforward extension of the basic behavior-based framework. Transitions are themselves behaviors that act to 'glue' two distinct high-level behaviors together. When a behavior transition notices that it is time to switch between two higher-level behaviors, it takes over from the old behavior. Instead of switching abruptly to the new behavior, it 'finishes up' for the old behavior and introduces a plausible transition to the new one.

The technical reader may now feel a burst of trepidation at the additional burden of work transitions may introduce. After all, if a transition is needed to connect any 2 behaviors, then for n behaviors we will be forced to write $O(n^2)$ transitions! We will see in Chapter 7 that while we will probably need to write at least $O(n)$ transitions, the actual number of transitions needed is limited, through mechanisms including their localization within high-level behaviors and their generalization (transitions that can go either from or to any arbitrary behavior).

Translation for non-technical readers: if we have to write a transition for every combination of 2 behaviors, then for 10 behaviors we need to write about 100 transitions, for 20 behaviors we need to write about 400 transitions, and so on. That is too much work to be practical. But it will turn out that the actual number of transitions needed is far less.

The ways in which transitions work and the architectural foundations they need are the subject of the rest of this chapter. We will start with a survey of the support for behavior blending that already exist as parts of various agent architectures. This will provide the basis for the architectural mechanisms — sign-management, transitions, and meta-level controls — that allow designers (1) to build agents with respect to how they will be interpreted, and (2) to use transitions to de-atomize those interpretations.

The Magic Principles

1. Don't integrate the agent; integrate the user's understanding of the agent.
2. Don't stop atomizing; change the choice of what to atomize. Let the designer understand and control the effect of the created agent.

Behavior Blending: State of the Art

In order to blend behaviors, we need to have techniques that allow us to combine behaviors together. In both classical and alternative AI, the technique most commonly used when two behaviors need to be combined is to interleave the agent's actions. For example, planning approaches for conjunctive goals integrate behavior by interleaving activities for each goal without any smoothing between them. The subsumption architecture, Pengi, and ANA all rely on interleaving actions to combine behaviors.

Here, we want to actually blend behaviors together. In order to find tools for this, we need to find ways in which you can combine behaviors that can smooth, average, or otherwise compromise between two behaviors, turning a discrete behavior break into a smooth transition from one to the other. While this smoothing has not previously been done on complex high-level behaviors, there are a number of techniques already available for smoothing between lower-level actions.

Ideas from Graphics

Blending is common, for example, in computer animation. In animation, it is clearly not appropriate for a character (or inanimate object, for that matter) to jump jerkily from pose to pose. At the same time, animation studios do not want to waste money and time by having highly-skilled animators draw all 24 frames a second in order to generate a half hour of animation. One of the common techniques to handle this is to use keyframes to specify a character's actions, and then use a process called "in-betweening" to provide smooth transitions from keyframe to keyframe. "In-betweeners" used to be humans, but they are now mostly replaceable by programs that can do the same thing. These programs can do various kinds of interpolation (averaging) between frames to smooth them out.

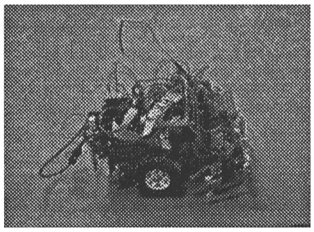
The related work section is essential for placing the developed technology in the context of a community. This section gives credit where credit is due for ideas that inspire the current work. It can be used for reference by people trying to do something similar to your work. It often also includes a component of explaining how your developed technology is different from (and by implication better than) what other people do. I managed to withstand this temptation here since I go so far as to devote several entire chapters to this argument elsewhere.
If you are not technically trained, this section may be hard to follow, since fully explaining each related technology for a non-technical audience would double the size of this already bloated thesis. Nevertheless, I would encourage you to hang in there and try to read this section at a high level, so you get some flavor of how these problems are thought about and some idea of how the technology I develop relates to AI technology in general.

Other graphics systems allow you to specify the animation by providing various key poses, and using physical simulation to figure out how the object should move between the poses. Jerks that remain at the low level can be worked out by a process called "time domain supersampling." With this technique, the computer system generates twice as many frames as necessary and then blurs between them instead of jumping from discrete state to discrete state. More details on these graphical approaches to transitions can be found in [Watt, 1993].

Ideas from Low-level Action

While these graphical techniques do not map directly to agent action, they introduce the idea that you can smooth between two discrete states by doing various kinds of averaging between them. This idea has been applied to agent action as well, resulting in various techniques that average between actions to create smooth transitions.

Ken Perlin, for example, has built a system representing a human dancer [Perlin, 1995] who can follow discrete commands ("rhumba," "walk," "run," etc.) while moving smoothly from one behavior to the next. The "actions" in Perlin's system represent joint angles (e.g. "move left knee 30 degrees"). Each behavior consists of a set of actions over time. When switching from one behavior to the next, the weight of the "finishing" behavior is gradually reduced to 0, and the weight of the "starting" behavior is simultaneously gradually increased to 1. To determine the actual actions that the dancer does, it multiplies the weight of the behavior by the magnitude of the action, so that the dancer's behavior is gradually, for example, less rhumba-esque and more like walking. Perlin also adds some additional constraints to make sure that the combined activity actually makes sense. Interestingly, Perlin also mentions the value of having smooth activity be visible to the user, while the programmer can think purely in terms of the atomized, discrete behaviors.


Steels's robot

Luc Steels' agent architecture, which is used to run robots, works on a similar principle [Steels, 1994]. Like Perlin, Steels explicitly states that smooth behavior switching is one of his goals. Steels criticizes the concept of action-selection as being incapable of generating smooth behavior because it implies jumping from action to action. Instead, Steels has all behaviors running all the time, with the resulting action commands being added together to generate the robot's final activity. For example, if one behavior wants to turn left, and one wants to turn right, the result will be that the robot goes straight ahead. Since this clearly could result in nonoptimal behavior (for example, if the robot wants to turn either left or right because there is a wall right in front of it), behaviors need to be developed hand-in-hand so that the additive principle works out correctly (rather than the independent behavioral development of many others in alternative AI).

Ideas from blending low-level actions for high-level behaviors

These systems focus on relatively low-level action (mostly moving around). Systems that are going to combine high-level behaviors will necessarily be more complex. These systems often have a motor-level component that is in charge of action (e.g. "go 3 feet to the left") with a high-level component that takes care of high-level behaviors (e.g. "go to the store") and sends orders to the motor system. In order for the action to look plausible, these systems, too, have various techniques for combining actions.

Blumberg's Hamsterdam [Blumberg, 1996], for example, has a motor level system which takes care of the low-level details of the agent's activity. An agent's body has various "Degrees of Freedom" that represent things an agent can move independently (for example, wagging its tail is usually independent of sticking its tongue out). Motor Skills are various low-level physical actions the creature can engage in that effect some of the agent's Degrees of Freedom, like "walking," "wagging tail," "putting ears back," etc. Motor Skills can be blended in two ways:

1. If Motor Skills affect different Degrees of Freedom they can happen simultaneously (you can walk and chew gum).
2. Consecutive Motor Skills can be smoothed by always putting the body in the same posture between the Skills. Silas the dog always stands up between actions; this makes sure that he doesn't, for example, spring from a lying-down behavior into a begging position. Unfortunately, this also means that he will stand up between lying-down and sitting down, which Blumberg points out doesn't seem quite right.

Both the Woggles [Loyall and Bates, 1993] and the Industrial Graveyard have a motor system that is at its most basic level surprisingly similar to Hamsterdam, given that they were developed separately. These agents have "body resources" which correspond to Hamsterdam's "Degrees of Freedoms." For these non-biologically-inspired agents body resources include such things as the bottom of the agent, the top of the agent, and the angle the agent is facing. Agents have a set of low-level physical actions that they can engage in; things like "squash," "spin," and "jump."

In these Hap-based systems, agents' actions are physically simulated. One benefit of this is that the graphics system that runs this simulation takes care to smooth the actions together appropriately. Between two consecutive actions, the system calculates an appropriate intermediate state based on the physics of the world. An agent, for example, that strings together two jumps will take care to land the first jump to transfer its momentum into the second; an agent that is jumping once and then stopping will land in a way to stop its momentum (otherwise it would fall on its face). This means that, unlike in Hamsterdam, there are no stereotypical in-between states the agents always engage in to move from one behavior to another.

One of the most complex and interesting methodologies for combining low-level action for high-level behaviors is explored by Gerald Payton and his colleagues. In both Hamsterdam and Hap, behaviors can ask for an action; if they conflict, the most important behavior's action actually happens, while the other behaviors have to wait until the important behavior is done. In Payton's system, behaviors give, not a single action command, but a range of preferences for various actions. The preferences of all behaviors are combined according to the importance of each behavior, and the best resulting action is selected. Behaviors can say both which actions they want, and which they don't want; Payton's system therefore avoids Steels's system's problem of the robot running into the wall, since both behaviors will say they do not want the robot to go straight. Additional details of the action specification mechanism can be found in [Payton *et al.*, 1992].

Ideas from high-level behaviors

It should be clear at this point that there already are a number of reasonable solutions to the problem of low-level behavior blending. There are several useful techniques for behavior blending, from various forms to averaging, to simultaneously engaging in both behaviors, to moving to set in-between states, to using physical simulation to determine how the actions can be combined properly. However, these techniques are not always appropriate for high-level behaviors. How can you average between going to the store and staying at home? Should an agent always stand stock-still, looking straight ahead, between any two high-level behaviors? Can physical simulation tell you how to move from dancing the rhumba to eating dinner? At some point as behaviors become more complex, the meaning of a behavior becomes more than the physical actions of which it consists (including, for example, groups of conditions under which different actions are appropriate). At this point simple averaging or weighting schemes no longer suffice to blend one behavior appropriately into the next one.

Clearly, the first step in blending behaviors is being able to blend the actions of one behavior into that of the next; for this we can use some of the techniques of the previous section. Now, we will take a look to see what support we currently have for blending together high-level behaviors themselves, and not just the actions they output.

There was no direct support for interbehavioral effects in the original version of Hap [Loyall and Bates, 1991]; the only tangential support was the availability of global memory.[2] The resultant difficulties in creating coherent behavior were noticed by both Bryan Loyall and Scott Neal Reilly, who add new mechanisms for interbehavioral support in their respective theses [Loyall, 1997a] [Neal Reilly, 1996]. Loyall adds dynamic variables, which allow different behaviors to share information about what they are doing; these variables can then be used by behaviors to coordinate what they are doing. A more direct support for behavioral

[2] Similarly, the subsumption architecture provides message passing, but as far as I know this is not used to support behavior blending.

coherence is provided by Neal Reilly, who introduces the concept of "behavioral features." A behavioral feature is an overall emotional attribute that the agent's behaviors should display (for example, "fear," "anger," "happiness"). Behavioral features are used in many behaviors to modify their action in order to create the overall impression of a coherent and identifiable emotional state.

Blumberg uses "Internal Variables" to allow his behaviors to share information in a way somewhat analogous to behavioral features. An Internal Variable is information that is local to a particular behavior, but will be shared with another one. For example, an Internal Variable could be "Focus of Attention;" behaviors sharing this variable will make sure the agent's activity, though switching from behavior to behavior, remains focused on the same object in the agent's environment.

Herman the Bug

An additional twist Hamsterdam makes is to allow behaviors to make different kinds of action commands, which can be blended in different ways. A behavior may issue a "primary" command, which basically means "do it!" A behavior that merely wants to make a recommendation can issue a "secondary" command, which means "do it unless someone more important objects." Or, a behavior can make a "meta-level" command, which means "if anyone wants to do it, they should do it this way" (e.g. "if I am going to walk, then it had better be slowly!"). This last kind of command can be used to create an effect like behavioral features, by getting the behaviors to generate a style of action that is coherent over the different behaviors that may control the body.

These systems add some tools into the behavior blending mix. The system that currently has by far the greatest level of blending and transition support, though, is Lester and Stone's Behavior Sequencing Engine [Stone, 1996] [Lester and Stone, 1997]. The "Behavior" in this title is something of a misnomer, since their system actually sequences not programmed behaviors but hand-made animation clips of their character, Herman the Bug. While some of their techniques are limited to sequencing clips, others can be generalized to more complex behaviors as well.

Herman is a pedagogical agent, whose role is to supervise students in an educational simulation, stepping in with advice when students seem to be getting lost. Because children are impatient with characters that are supposed to be alive but seem wooden and mechanical, Lester and Stone's system is specifically focused on generating *visually coherent* activity for their agent. At the low level, film clips are sequenced seamlessly by using a technique called "visual bookending." This means that the start and end frame of each clip is chosen from a small set of possible "home" frames, and only clips with the same home frame are sequenced together. This system is analogous to Silas's movement to standing between his behaviors, although the use of a variety of "in-between" states reduces the danger of stereotypicity. If clips that must be sequenced have different keyframes, a transition animation is played in between to move from one to the next.

At a higher level, since Herman spends a lot of time explaining concepts, much attention is paid to making these explanations coherent. Rather than jumping from topic to topic, Herman uses 'topical transitions' between different explanatory behaviors. In addition, when Herman has been quiet and now wants to launch into a very noticeable behavior, he uses an anticipatory action to alert the student he is about to do something he or she should notice. For example, if he has been lying down, he will sit up before he launches into an explanation.

At this point, we have various tools in various frameworks for supporting behavior blending. Each of these provides part of the answer. Turning this into an adequate AI technology requires a few more pieces:

1. We need to have some conception of the full range of kinds of transitions, so we have some idea of what the architecture needs to support.
2. We need a common framework that will allow us to provide support for these different kinds of transitions in a single system.

This is the goal of the rest of the chapter.

Design of the Expressivator

The architecture designed here is called the *Expressivator*,[3] since, unlike most current systems, it focuses on the ways in which the agent expresses its designer's intentions to the audience, rather than on what the agent is doing internally from moment to moment. The goal of the technology developed in this chapter is to be able to de-atomize the agent from the user's point of view, by introducing techniques for smoothing between observed behaviors using transitions. This will involve three major components:

1. We need to provide the agent author with a way of being able to program the agent with respect to what the user sees the agent do (not just what the designer thinks the agent is doing). Agents built under the Expressivator are structured using *signifiers*, which are behaviors that are explicitly communicated to the audience through the use of low-level *signs*. The agent uses a *sign-management system* to keep track of signs and signifiers that have been communicated, allowing it then to decide what to do based not only on sensing and its internal state, but also on what has been communicated to the user.
2. We need to get some idea of the range of possible kinds of behavior-blending *transitions*, so that we have some idea of the kinds of things the architecture needs to support. These transition types specify different ways in which high-level behaviors can be smoothed together.
3. We need to add structures to the architecture that will allow it to support this range of transition types. The Expressivator does this through the use of *meta-level controls*, or special mechanisms which transition behaviors can use in order to sense and alter the behaviors they connect. In addition to supporting transitions, these controls allow the agent designer to explicitly coordinate and communicate the relationships between behaviors, rather than leaving the coordination of behavior as an implicit property of the agent architecture.

[3] Yes, this name *is* supposed to evoke images of 60's optimistic futuristic culture.

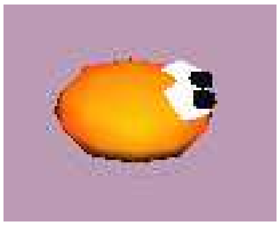


FIGURE 5.7: A Woggle that is 'clearly' moping.

These pieces together will form the structure of the Expressivator.

Signs, Signifiers, and Sign Management

As mentioned in Chapter 2, in 1992, a group including many members of the Oz Project built the Edge of Intention, a system containing small, social, emotional agents called Woggles that interact with each other and with the user. While building the agents, we took care to include a wide variety of behavior, which ranged from simple behavior like sighing and moping to relatively complex social behavior like follow-the-leader and fighting. At the same time, we made sure that the agents did not blindly follow the user but had a 'life of their own;' we hoped that this would make them more compelling personalities to get to know.

At the time, we believed that the individual behaviors of the agents were reasonably clear. After all, we — their builders — could usually tell what they were doing ("A-ha! It's small and flat! That means it is moping!" — see Figure 5.7 and judge for yourself). Soon, however, we found that it was difficult for other people to be able to understand the behaviors and emotions we were trying to communicate through the Woggles. Users were at a disadvantage because, unlike us, they did not actually have the code memorized while they were watching the agents. Because we — the builders — thought in terms of the underlying behavior names in the code, we had thought the agents' behavior was clear. This had led us to neglect to some extent the *external* behavior of the agents. Behaviors were not always programmed with enough observable actions that the audience could actually tell what the agent was doing.

For de-atomizing the user's impression, allowing the designer to control the impact, the external appearance, of the behavior is extremely important. But our lack in this department was not due (merely) to a perverse attitude about how agents should be programmed. Most current behavior-based architectures only allow designers to write code more or less purely based on an internalistic perspective. Agents make decisions based on what they perceive, on what they have recently done, or on their current mood — but not based on *what the user thinks the agent is doing*. This is partly a consequence of the attitude described in Chapter 4 that agents are fundamentally autonomous problem-solvers, and that therefore any impression the user may have of them irrelevant. But it is also partly a consequence of the difficulty of figuring out what on earth the user is thinking.

For many computer scientists, there are two main strategies that immediately suggest themselves for figuring out what is going on with the user: (1) perceive what the user is doing and try to figure out from that what the user is thinking; or (2) make a general model of what a typical user would be thinking, and use that to predict what the current user thinks of the agent. Neither of these options is in itself particularly compatible with behavior-based AI, since they both require a substantial amount of modeling and reasoning. Chances are also fairly good that either approach will be wrong a lot of the time — mind-reading is not well-developed among humans, let alone among computers.

Believable Agents research suggests that there is a third way out. In this view, there *is* a way to give the designer access to the presentation of the agent as comprehended without having to model or perceive the user, and that is by turning the tables on the user. The user could really be thinking anything; but the *designer* knows what he or she *wants* the user to think. The goal, then, is not to have the agent try to figure out what the user thinks, but rather to provide the designer with support for communicating as clearly as possible through the agent what the user should be picking up. Since designers are generally much more savvy about cues a human observer might pick up than an agent can be, this puts the most competent agency in the driver's seat.[4]

Non-technical readers may recognize this strategy from the arts. Directors of films, composers of music, and authors of books (and technical reports, for that matter) also do not know exactly what the 'user' of their works is going to pick up on, but they generally do not feel the need to develop a scientific, testable model to find out what the observer is thinking. Rather, they rely on their intuition, a tradition of techniques, trying things out on themselves, friends, and test audiences, and a preoccupation with presentation in order to communicate their concept successfully to the audience. The argument Believable Agents researchers make is merely that these sorts of things can also be tapped for AI.

Agent Structure for Communication

The goal of sign management is to provide support for communication within the agent design. The Expressivator implements sign management through the following three mechanisms:

- The agent's low-level activities are structured into *signs*, which communicate the meaning of the agent's actions directly to the user.
- The agent's high-level activities are structured into *signifiers*, i.e. behaviors *which are explicitly intended to be communicated to the audience*.
- The *sign-management system* keeps track of the signs and signifiers that have been communicated to the audience. Signs and signifiers are posted to memory when they have been communicated. This allows the agent to base its activity, not only on what it sees around it and where it is in its internal code, but also on *what the user has seen the agent do*.

[4]This philosophy is similar to the "Inverse User Model" suggested by Michael Mateas [Mateas, 1997] to manipulate users into 'seeing' the world in an author-chosen way.

At a high level, the motivation for sign management can be understood in the following manner. Typically in alternative AI, the behaviors with which the agent is programmed are activities which allow the agent to achieve its goals. Here, behaviors are better thought of as 'activities to be communicated to the user.' The Expressivator therefore structures agents according to *levels of meaning-generation*. Behaviors are not problem-solving units, but units of meaning to be communicated to the user, and they are organized according to the kind of meaning they communicate.

In Hap, for instance, behaviors are at the most fundamental level designed out of physical actions — such as "jump," "squash," or "spin" — and mental actions — such as "calculate a good angle for me to face".[5] Actions are combined into low-level behaviors, such as "say hi," "watch out for insults," or "walk to bed," which are small units of useful behavior. These units are then combined into high-level behaviors, such as "play follow-the-leader," "have a fight," or "take a nap," which represent what the agent is basically doing. The lines between low-level and high-level behaviors are not clearly drawn, but they provide a useful framework for thinking about behavior design.

In the Expressivator, the fundamental units of behavioral design are not physical actions that have effects in the world, but *signs* that have effects on the user. Signs, physical actions, and mental actions can be combined to form *low-level signifiers*; these are behaviors, which are differentiated from low-level behaviors only in that they are explicitly intended to be recognized by the user. Low-level signifiers can in turn be combined into *high-level signifiers*, which are behaviors which communicate the fundamental activities the user should be able to recognize in the agent. A *sign-management system* keeps track of when each sign and signifier has been communicated to the user. Now, we will take a look at each of these mechanisms in more detail.

Signs

The most basic unit of agent structure for most behavior-based architectures is also the most basic unit of physical activity, the physical action. Physical actions are commands to the motor system like "move hand left," "raise head," etc. While the Expressivator certainly composes behaviors out of physical actions, the design of the agent is not so much focused on what the agent is physically doing, but *how the agent's action will be interpreted*. This means that, at the design level, the most basic unit through which an agent is structured for the Expressivator is not the physical action but the *sign*.

A sign is a token the system produces after having engaged in physical behavior that is likely to be interpreted in a particular way. This token includes an arbitrary label (like "sigh") that is meaningful to the designer, and represents how the designer expects the physical behavior (like "stretch up for 100 milliseconds and then squash down for 100 milliseconds") will be interpreted. This token is stored by the sign-management system, so that the agent can use it to influence its subsequent behavioral decisions.

[5] Mental actions are expressed in C or Lisp code.

Behavior: *Harass patient to follow scheduled activity*

1. Go to schedule
2. Read schedule
3. Look at clock
4. Look at schedule
5. Look at patient
6. Wait a moment for patient to comply
7. Look at schedule
8. Look at patient
9. Shake head
10. Approach patient menacingly ....

FIGURE 5.8: Example of a behavior and its signs

Formally, a sign is an arbitrary label (such as "saw possible insult") and an optional set of arguments that give more information about the sign (such as "would-be insulter is Wilma"). A behavior can 'post' a sign each time it has engaged in some physical actions that express that sign, using the `post_sign` language mechanism. For example, after moving its head slowly from left to right, the agent may post a sign "read_line" with an argument of the number of the line it just 'read.'

Figure 5.8 shows an example of a behavior and the signs that are emitted during it. At first glance, these signs look like low-level physical actions, but there are important differences. Rather than corresponding to simple movements an agent can engage in, a sign corresponds to a *set* of such movements that *carries meaning to a user*. The "reading" sign, for example, combines a set of low-level actions as the lamp's head moves from left to right across each line of the schedule. More fundamentally, signs are different from both actions and traditional behaviors in that they focus on *what the user is likely to interpret*, rather than what the agent is 'actually' doing. When "reading," for example, the agent does not actually read the schedule at all (the locations of the lines and their contents are preprogrammed); it merely needs to give the *appearance* of reading.

Figure 5.9 shows how the `post_sign` language construct is used while the agent is walking; after each step, it posts that the user has seen it take a step towards a particular goal point.[6] Signs are context-dependent in the sense that the designer notes the meaning of physical actions within the context of the behavior in which the action appears. This means that the same physical actions might result in quite different

[6] It needs to keep posting the sign, even after the first step, in case the behavior is interrupted.

```
(sequential_production walk_towards (gx gy)
  (with (success_test
         (... agent has reached goal point ....))
     (with persistent
       (seq
         (subgoal take_step_to $$gx $$gy)
         (post_sign walking_to
                    ((x $$gx) (y $$gy)))))))
```

FIGURE 5.9: The 'walk_towards' behavior and the sign ('walking_to') it posts.

signs, depending on context: for the lamp, while walking, jumping to a new spot results in a "taking a step" sign, while during headbanging, the same physical action leads to a "hop around" sign.

Signifiers

Physical actions, mental actions, and signs are combined into low-level signifiers. Signifiers are behaviors that are *explicitly intended to be communicated*. Low-level signifiers correspond to low-level behaviors; they are relatively simple behaviors that convey a particular kind of activity to the user. In the Industrial Graveyard low-level signifiers include things like "hit head on ground," "tremble and watch the Overseer," "look around," and "go to an interesting spot." Low-level signifiers differ from low-level behaviors in ordinary behavior-based architectures in that users should be able to identify the low-level signifiers more or less correctly — which is otherwise not necessarily the case.

For example, a low-level behavior for the Woggles might be "watch out for insults." This behavior would consist mainly of sensing to make sure that no one is coming nearby and engaging in the "In Your Face" activity, which is the highest insult one Woggle can pay another. This sensing, however, does not have any component that is visible to a user. There is no way for the user to know that the agent is trying to avoid being insulted — the only way for the user to get this idea is to see the agent being insulted, watch it react, and then hypothesize that the agent was watching out for insults all along.

Turning "watch out for insults" into a low-level signifier means adding signs to it that communicate what the agent is doing to the user. An agent watching out for insults in this sense might glance around now and then, becoming nervous when it notices a frequent insulter coming nearby.[7] Now the user knows that the agent is paying attention for something — and, incidentally, is not caught off guard when the Woggle goes into a state of frenzy upon finally actually being insulted.

Low-level signifiers are identified by marking behaviors when invoked. This is done using a special marker, `low_level_signifying`, which has been added to the behavior language. The behavior 'smack_head' would be invoked as `(subgoal smack_head)`; to make it a low-level

[7] A Woggle might do these things too, but they will be components of other behaviors that are coincidentally displayed, not part of the watching for insults behavior itself.

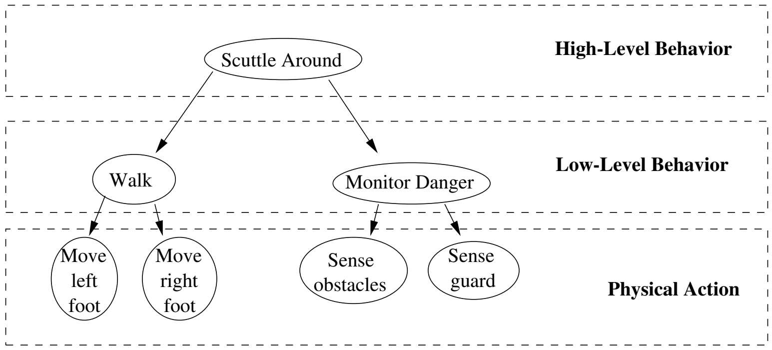


FIGURE 5.10: A typical behavior structure divides the agent into objective units of activity.

signifier, it is invoked as `(with low_level_signifying (subgoal smack_head))`.

Low-level signifiers can be combined to build up *high-level signifiers*. High-level signifiers are, like low-level signifiers, behaviors that are intended to communicate the agent's activity. High-level signifers are composed of low-level signifiers to form a complex, high-level activity. The Industrial Graveyard includes high-level signifiers like "head-banging," "exercise," and "be killed," These, like low-level signifiers, are intended to be communicated to the user. The rule of thumb is that low-level signifiers are groups of actions that can be grasped and comprehended as what the agent is doing on a moment-by-moment basis. High-level behaviors are what the agent should be thought of as doing at a whole. They extend over time and are composed of various low-level behaviors, which they organize into an intentional unit. The high-level signifiers, in turn, combine to form the complete activity of the agent.

High-level signifiers are identified in the analogous manner to low-level signifiers. A special marker, `high_level_signifying`, is added to the language. The 'headbanging' high-level signifier, for example, can then be invoked this way: `(with high_level_signifying (subgoal headbanging))`.

Summary: Signs and Signifiers

To summarize, a typical behavior-based architecture structures the agent according to its objectively determinable activities. To build a behavior like "scuttle around," in which the Patient wanders around the graveyard while trying to avoid danger, the high-level "scuttle" behavior may be broken into low-level walking-around and danger-sensing behaviors, which are in turn broken up into the physical actions (including sensing) of which they are composed (Figure 5.10). In the Expressivator, on the other hand, "scuttle-around" is a high-level signifier, which is broken into low-level signifiers, which are then broken into signs (Figure 5.11). Because signifiers and signs are explicitly intended to be communicated, the structure of the agent may change; for example, instead of simply sensing danger, the Patient actually moves its head and eyes around to look for danger, to be sure that the user will know what it is sensing and

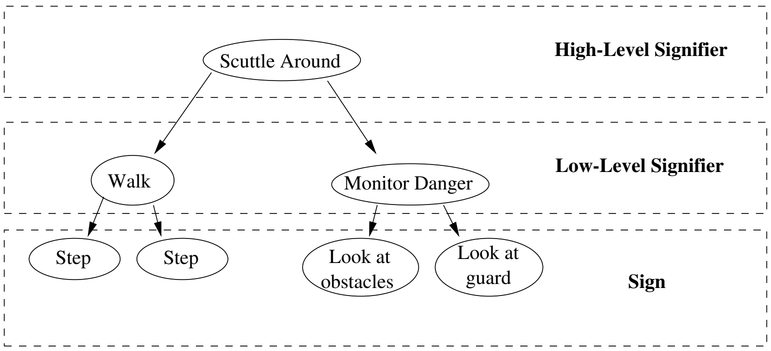


FIGURE 5.11: The Expressivator behavior structure divides the agent into subjective units of meaning to be communicated.

be able to identify the "monitor danger" behavior. In this sense, signs and signifiers help the designer to design the agent so that the designer's chosen behaviors actually are communicated to the user.

Sign Management

So far, I have described signs and signifiers in terms of how the designer can use them to structure their agent with respect to eventual user interpretation. But it would also be nice if the agent itself could reason about how the user is currently interpreting it. For example, if the agent is about to walk across the world, but the user most recently saw it hiding from the Overseer, the agent can modify its walking behavior to include glances at the Overseer so that the change in behavior seems less jarring. The *sign-management system* helps the agent to keep track of the user's current likely interpretation, so what the user is likely to be thinking can influence behavioral decisions in the same way as environmental sensing and internal state do.

The most obvious way for the agent to keep track of what the user thinks is for it simply to notice which signs and signifiers are currently running. After all, signifiers represent what is being communicated to the user. But it turns out in practice that this is not correct *because the user's interpretation of signs and signifiers lags behind the agent's engagement in them.* That is to say, if the agent is currently running a "headbanging" signifier, the user will need to see the agent smack its head a few times before realizing that that, in fact, is what the agent is doing.

The sign-management system deals with this problem by having the agent *post* signs and signifiers when it believes the user must have seen them. As mentioned about, the `post_sign` language construct is used to remember that a particular sign has been displayed. Similarly, the `post_low_level_signifier` and `post_high_level_signifier` constructs are used to remember that particular signifiers have been displayed. The question, then, is how the agent knows when the sign or signifier has been displayed and can therefore be posted.

Signs have been displayed — and are therefore posted — whenever the agent has done some physical activity that expresses the sign (Figure 5.12). "Posting" means the agent stores the sign and its arguments

```
(sequential_production smack_head_emotionally ()
  (locals (time "random_range(350,800)"))
  (par
   (subgoal snap_head $$time)
   (subgoal swing_head $$time)
   (subgoal squish_body $$time
            "random_range(-10,10)"))
  (par
   (act "ASquashHold" 0 "$$time / 2")
   (act "AElevateEyesTo" 0 "$$time / 2"))
  (post_sign smack_head_once))
```

FIGURE 5.12: Signs are posted once their physical actions have been engaged in.

```
(parallel_production hit_head ()
  ....
  (with effect_only
      (demon (("G (Goal CurrentSign
                          == slap_head_once;);"))
 (post_low_level_signifier hit_head)))
```

FIGURE 5.13: Low-level signifiers are posted after a demon notices that appropriate signs have been posted.

in memory; the agent also notes the time the sign was expressed. Now, other behaviors that want to know what the agent has been doing from the perspective of the user can check memory to see which sign has recently been posted.

Low-level signifiers, in turn, can be assumed to have been displayed when some key signs have been emitted. They therefore watch the signs that are emitted to find out when they have been expressed (Figure 5.13). For example, “look around scared” watches for a “scared glance” sign to be posted. When the “scared glance” sign appears in memory, the agent can start having some confidence that “look around scared” is starting to be communicated, too. The agent then posts that “look around scared” is being communicated, using a mechanism analogous to posting signs. In general, when the right signs have been posted, low-level signifiers post themselves, in effect announcing that the user should have seen them, too.

High-level signifiers, in turn, have probably been displayed when key low-level signifiers are expressed. They therefore watch for the posting of their low-level signifiers. When the right combination of low-level signifiers have been posted, the high-level signifier is posted as well (Figure 5.14). In this way, the agent can keep track of the impression it is making on the user, from the details of signs to the overall impression of high-level behaviors. More technical details of how this works can be found in section A.1 of the Appendix.

Once signs and signifiers have been posted, other behaviors can check

```
(parallel_production head_banging ()
  ....
  (with effect_only
      (demon
         (("G (Goal CurrentLowLevelSignifier
                       == hit_head;);"))
         (post_high_level_signifier head_banging)))
}
```

FIGURE 5.14: High-level signifiers are posted after a demon notices that appropriate low-level signifiers have been posted.

to see what has been posted recently before they decide what to do. Behaviors can check for arbitrary sequences of signs and signifiers. The end result is that *the signs and signifiers the agent has expressed can be used just like environmental stimuli and internal drives to affect subsequent behavior*. This means that in the Expressivator, behavioral effects on the user have the same status as action memory and perception in other systems. For example, a *watch-guard* behavior may check recent signs and notice that a *hide-from-guard* sign was posted; in this case, it would know to maintain behavioral coherence by peering at the guard carefully, rather than walking right up to the guard to see what it is doing.

Summary of signs, signifiers, and sign management

One nice property of this hierarchy of meaning-production is that it follows our principle of maintaining modularization in order to simplify agent design. Signs, low-level signifiers, and high-level signifiers can still be designed separately. When combining them into the full system, each level only needs to worry about the level directly under it. Signs only need to be concerned with the physical actions that express them; low-level signifiers only care about signs, not about physical actions; and high-level signifiers only need to worry about low-level signifiers, not signs.

Signs, signifiers, and sign management also provide the basic support for our other principle, i.e. designing agents with respect to their impact. In fact, the sign-management system improves not only the impact of the agent's behavior but also that of the agent-builder's! This is because, in addition to allowing agents to reason about what the user sees, it also forces the *designer* to reason about those things. By noting every time a sign or signifier is supposed to have been communicated by a behavior, builders' attention is focused on the problem of breaking a behavior into signs and signifiers and then making sure that they are expressed. The structure of the sign-management system encourages them to think about behavior in terms of signs and signifiers, and to construct appropriately expressive low-level behaviors to display those signs and signifiers.

Behavior Transition Types

Sign management provides the foundation for de-atomizing the agent's impact, since it allows us to design the agent with respect to its probable interpretation by the user. With this under our belt, we can turn our attention to providing support for behavior blending. The first step is to try to get a handle on the range of possible ways that behaviors can be combined. In this section, we will look at a variety of ways in which this can be done.

Generality is a great virtue in Artificial Intelligence (as in other sciences). Even researchers whose goal is to construct technology that is specific to a particular environment want to give the general rules of specificity. Ian Horswill gives an elegant example of how to do this. He builds an architecture radically specific to an environment — and then shows exactly which parts are specific to which properties of the environment, and therefore need to be replaced for the robot to run in another environment [Horswill, 1993]. In this section, then, I try to get as broad an idea of transition types as possible so that the Expressivator can be built to support as many of them as possible.
It may seem to you that I am basically making most of the stuff in this section up. If so, it is because I am. This is basically a form of brain-storming based on current architectures and on where they could go with the ideas of socially situated AI. I have no proof that this section is comprehensive, and I rather doubt that it is. But it is at least a place to start.

The analysis of already existing support for behavior blending suggests a number of transition types as a starting point:[8]

- *Parallel Behavior Blend*: Both Hap and Hamsterdam allow two behaviors to run simultaneously, sharing control of the agent's body. This is a meaningful form of blending when the behaviors use non-conflicting body resources (e.g. walking and talking).
- *Virtual Behavior Blend*: The subsumption architecture allows two behaviors to run simultaneously, while disabling one behavior's muscle commands. This means the disabled behavior can still perceive the world and influence the creature's emotions, but cannot move the agent's muscles. (It might be an interesting variation to allow the disabled behavior only to move the eyes; this way, the focus of attention of the disabled behavior can still peek through).
- *Average Behavior Blend*: The architectures for low-level action suggest that an interesting way of combining behaviors may be to average their action commands. It remains to be seen if this technique is meaningful for high-level behaviors.
- *Interruption*: If an agent intends to engage in a behavior for a very short time, it may make sense to merely interrupt one behavior with the other, then return to the first behavior when the second has completed. This is supported by nearly all current architectures, including Hap.
- *Sudden Break*: At times, the most appropriate way to combine behaviors is to jump from one behavior to another without transition. This can communicate that something sudden has happened to force the agent to switch rapidly, or that the agent has a highly reactive personality. You may have already noticed that this is the default in nearly all architectures — it is the definition of schizophrenia. But just because it is not so good to have sudden breaks *all* the time, this does not mean that it is *never* the right policy.

The example of the guard dog earlier suggests that one function of the transition is to make the reason for the switch to the new behavior plausible to the user. This means an important novel kind of transition can be the *Explanatory Changeover*. This transition is the default transition proposed when I introduced the concept of transitions on page 106: finish up the old behavior, engage in a sequence of actions that explains why the new behavior is being started, then start the second behavior.

[8]These categories were also inspired by my analysis of Luxo, Jr., which appears in Intermezzo II. While rhetorically it made sense to present them in this order, in practice the development of the ideas in this thesis was never so linear.

Finally, the *Accidental Transition* turns the Explanatory Changeover on its head by watching for and capitalizing on what the user would find plausible. The agent watches its recent behaviors for patterns of behavior that might seem reasonable to the user. When a particular pattern is launched, the agent can 'switch gears' to follow that pattern, instead of whatever it was planning to do originally. This is a technique frequently used by my cat in moments of embarrassment: rather than admit that falling off the windowsill was an accident, he finds some way of recovering so that it looks like he meant to do it all along. It is also akin to something that can be observed in split-brain patients, who manage with one side of the brain to spin narratives (albeit patently false ones) that structure actions taken by the other side.

I had gotten to this point in my analysis of transition types when I noticed there was something strange at work. Even though I had repeated my magic mantra of de-schizophrenization hundreds of times, I still found myself slipping back into my straightforward, technical, agent-as-autonomous mindset. Perhaps you have noticed the flaw in this line of reasoning already: *all the behavior transition types mentioned so far work with respect to the agent's internally-defined behaviors,* not *with respect to what the user sees*. The real question is not how behaviors can be combined, but how the user can be given the *impression* that behaviors are being combined. It turns out this re-formulation can make the problem much simpler — by avoiding the complexity of actually having two full-blown behaviors running simultaneously.

With this lesson firmly ingrained (or so I thought — the Doctrine of Agent Autonomy turns out not to be so easily erased from an AI researcher's world view), I went on to design several 'impressionistic' transition types:

- *Subroutine Behavior Blend*: Don't run both behaviors simultaneously; rather, take some 'representative' subbehaviors of one behavior and combine them with the other behavior. The idea here is to still give the user the 'flavor' of the behavior, without actually having the complexity of doing both behaviors simultaneously.

- *Principled Subroutine Behavior Blend*: Why stop at reducing only one behavior? Pick just a few subbehaviors of both behaviors, and combine them in a single blended behavior. This has the advantage of letting you weed out the subbehaviors of the 'dominant' behavior that conflict with the subbehaviors you would like to add to it.

- *Symbolic Reduction*: When it comes down to it, you don't even need to use *any* of the subbehaviors of the first behavior. Rather, the behavior can be reduced to a simple symbol or sign — a tick, a focus of attention, a particularly poignant movement — that is easy to incorporate in the second behavior. Note that this is similar to the use of Internal Variables in Hamsterdam, though with a different goal.

- *Reductive Behavior Blend*: We can make things yet simpler again. The Reductive Behavior Blend reduces the first behavior to an attribute whose value can vary — "mope" can be reduced to slowness; "hide from Overseer" can be reduced to fear; "escape from Overseer" can be reduced to agitation. This attribute is then used to modify the second behavior. We can now combine two behaviors with various emphases between them simply by varying this attribute's value from 0 to 1.

- *Off-screen Transition*: Since the goal is to blend the user's impression of the agent's behaviors, if the user is not looking at the agent at all, the agent can simply jump from one to the next.

- *Unknown Transition*: Sometimes, none of the agent's behaviors are appropriate. Rather than sitting around like a lump of silicon, the agent should fill in the 'lulls' between behaviors. A good way of doing this is to add a behavior that merely looks around the world or at the most recent object of attention.

Taken together, these 12 transition types almost certainly do not tell the full story of all the ways in which behaviors can be combined. They do, however, provide the groundwork for the kinds of ways of combining behaviors that the Expressivator should support. In the next section, I will introduce *meta-level controls* as a way to support these transition types — in addition to providing a form of de-atomization themselves.

Meta-Level Controls

At this point, the Expressivator is equipped with techniques for designing agents' impressions, and we have some idea of the kinds of transitions we would like the Expressivator to support. Now all we need to do is actually implement them.

It turns out that this is not entirely straight-forward. Most transition types depend on the agents' behaviors to know about and coordinate with one another. However, most behavior-based architectures are based on the idea that behaviors should be shielded as much as possible from one another. Because behaviors engage in minimal communication, it is difficult for behaviors to know enough about each other to coordinate.

There are good reasons for this kind of black-boxing. Making behaviors highly interrelated makes them harder to program, and makes it harder to add new behaviors to an already-built system. The image Brooks produces of being able to add new behaviors without making any changes to the old system is therefore highly attractive.

The question that faces us, then, is the following: *what is the minimum amount of de-modularization we can do and still have behavior blending work*? We will investigate this question by finding a small set of meta-level controls that will support the full range of behavior transition types listed here. It will turn out that, with the exception of the average behavior blend, the set we need is small, reasonable to implement, and useful for things besides transitions, as well. In particular, it will turn out that meta-level controls add to the expressiveness of behavior-based architectures in ways that will turn out to be crucial in Chapter 7 — by making explicit, and therefore expressible, the formerly implicit interactions between behaviors.

Meta-level controls to implement transitions

Transitions at their most basic work as glue between an old behavior and a new behavior. Generally, they need to know when the old behavior needs to be terminated, delete the old behavior, engage in some action, and then start the new behavior. This means, at a minimum, that transition behaviors need to have all the abilities of a regular behavior, and a few more: (1) they need to be able to know what other behaviors are running; (2) they need to be able to delete an old behavior; and (3) they need to be able to begin a new behavior.

These abilities to know about and affect other behaviors I call *meta-level controls*. Because meta-level controls are explicitly intended for communication and coordination between behaviors, they are in some sense a violation of the behavior-based principle of minimal behavioral interaction. Nevertheless, meta-level controls are so useful for coordinating behavior that several have already found a home in behavior-based architectures. An example is Hamsterdam's meta-level commands, which allow non-active behaviors to suggest actions for the currently dominant behavior to do on the side.

The Expressivator attempts to systematize this use of meta-level controls. The goal for the Expressivator is to find a small set of meta-level controls that will support the full range of transition types. This set of meta-level controls, then, provides a common framework under which transition types can be implemented and combined.

A stroll through the behavior transition types reveals the meta-level controls sufficient to implement all these transition types:

- *Parallel behavior blend*: The behaviors run simultaneously. This needs no meta-level controls. It is currently supported by behavior-based architectures.

- *Average behavior blend*: For the average behavior blend to work, all physical actions need to be averaged before they are sent to the agent's body. This requires re-routing the action commands that behaviors make through the transition behavior, which then averages them before sending them to the body.

- *Subroutine behavior blend*: The transition adds a subroutine to an already-running behavior. Transitions need to have the power to take some subbehaviors and add them to other behaviors.

- *Virtual behavior blend*: Transitions 'paralyze' one of the two behaviors being combined. Transitions need to be able to turn off muscles of a particular behavior.

- *Reductive behavior blend*: Transitions need to be able to change the internal variables that affect how other behaviors are processed in order to make one behavior reflect the addition of another.

- *Symbolic reduction*: The transition adds a subroutine to express a simple version of a behavior another already-running behavior. This can be done using the same techniques as subroutine behavior blend.

- *Principled subroutine behavior blend*: The transition makes a new behavior by combining either already-running or new subbehaviors. Transitions need to be able to construct new behaviors out of subpieces that already exist.
- *Interruption*: This is do-able in current agent architectures.
- *Sudden break*: This is also do-able in current agent architectures.
- *Off-screen transition*: This needs no special powers — the transition only needs to know if the agent is visible.
- *Accidental transition*: Transitions need to have access to a memory of previous behaviors and to be able to match patterns of behaviors against it.
- *Explanatory transition*: As above; delete the old behavior, do some action, and start the new behavior.
- *Unknown transition*: Transitions need to be able to tell that there are no other behaviors active, and fill this time in with default behavior.

Summing these needed controls up gives us a complete set of meta-level controls, which will allow transitions to be built on top of almost any behavior-based architecture. Transition behaviors need to be able to do the following:

1. to *query* which other behaviors have recently happened or are currently active,
2. to *delete* other behaviors,
3. to *add* new behaviors, not as subbehaviors of the transition, but at the top level of the agent,
4. to *add new sub-behaviors* to *other* behaviors,
5. to *change the internal variables* that affect the way in which other behaviors are processed (I call these "Communicative Features"),
6. to *turn off* a behavior's ability to send motor commands, and
7. to *move running subbehaviors* from one behavior to another.

The average behavior blend might be easy to implement in an architecture like Payton's or Perlin's that supported action blending. It turned out to be nearly impossible to do in Hap because of the way Hap divides action implementation (the level at which averaging should happen) from behaviors (the level at which the transition should be able to invoke the averaging). The more I thought about this transition type, though, the less sense it made to me. How often does it make semantic sense to combine high-level behaviors like "eat" and "sleep" by averaging their muscle commands? It is possible that someone more creative than me will come up with a good use for the average behavior blend, but on the surface it did not seem to warrant a great deal of architectural effort.

The implementation of these meta-level controls in the Expressivator and their relationships with other schemes for meta-level reasoning is discussed in more detail in section A.2 of the Appendix. This is a must-read for the technically oriented, but I did not want to torment the humanists any more than necessary.

Meta-Level Controls In General

I originally foresaw meta-level controls purely as a way to implement behavior transitions. It turns out, however, that they have interesting properties in themselves. Most fundamentally, meta-level controls provide support for building expressive, communicative agents because they make explicit — and therefore expressible — parts of the agent that were formerly implicit in the architecture.

Specifically, most behavior based systems treat individual behaviors as distinct entities which do not have access to each other. Conflicts and influences between behaviors are not handled by behaviors themselves but by underlying mechanisms within the architecture. Expressing the reasons for the behavioral decisions the agent has made is difficult, when, for instance, the agent decides what to do by reducing behavioral appropriateness to a number and then choosing the behavior with the highest numerical value. In these cases, the designer may not even be able to articulate why the agent does what it does, let alone the agent itself. *Because the mechanisms by which the agent decides what to do are part of the implicit architecture of the agent, they are not directly expressible to the user*.

Meta-level controls make the relationships between behaviors explicit, just as much a part of the agent design as the behaviors themselves. They allow behaviors, when necessary, to affect one another directly, rather than having inter-behavior effects be subtle side-effects of the agent design. Meta-level controls give the agent builder more power to expose the inner workings of the agent by letting them access and therefore express aspects of behavior processing that other systems leave implicit. Behaviors in this framework can check on and coordinate with each other, increasing their ability to create a coherent impression on the user.

Putting It All Together: The Expressivator In Action

Now that we have sign management and meta-level controls, behavior transitions between user-identified behaviors become easy to write. Here, I will give some examples of how transitions are implemented in the Patient of the Industrial Graveyard, to give a flavor for how the architecture is used in practice.

Single Transition Type

When the Patient is in trouble, the Overseer comes over to 'administer meds.' It does this by striking the Patient on the head, which causes it to collapse and turn off for a period of time. This is a virtual behavior blend, which is implemented as shown in the pseudo-code in Figure 5.15. The virtual behavior blend uses the 'paralyzing' meta-level control in order to allow emotional processing (particular with respect to fear of the Overseer) to continue, while overriding muscle commands so that

Transition behavior: *Anything to turned off*
Precondition: *Patient is struck by Overseer*

1. **Paralyze** all behaviors
2. Close eyes
3. Find a direction in which I can fall down without hitting anything
4. Collapse in that direction
5. **Add the 'turned_off' high-level behavior**

FIGURE 5.15: Example of a virtual behavior blend (meta-level controls are in bold)

the Patient appears passed out. The 'adding new behavior' meta-level control is used to start the 'turned_off' behavior when the transition is complete.

Combining Transition Types

In practice, I often found it useful to *combine* transition types. In my experience, meta-level controls provide a flexible framework in which those types can be combined to produce whatever transition makes the most sense for the current behavioral change. For example, the Patient has a 'reading' behavior, in which it appears to be reading the daily schedule of events in the Junkyard, and an 'exercise' behavior, in which it does aerobics. When the Patient is reading the schedule during exercise time and the Overseer menacingly approaches, the Patient should switch from reading to exercising. Rather than switching abruptly, the Patient shows its reaction to the Overseer and switches to a panicking version of exercising. As the Overseer goes away, the Patient calms down and the exercise behavior reverts to normal.

This is implemented using a mixture of meta-level controls as shown in Figure 5.16. This transition combines an explanatory changeover (the Patient is switching because it notices the Overseer) with a symbolic reduction (the shock of being caught by the Overseer is reduced to the gesture of looking at the Overseer) and a reductive behavior blend (the exercise behavior is modified by the "energy" Communicative Feature which is at first set high to reflect the Patient's shock at being caught reading by the Overseer, then diminished as the Overseer leaves).

The Story So Far

In this chapter, we have looked at transitions as a form of de-atomizing the user's perception of the agent. I introduced the idea of structuring

Transition behavior: *Reading to exercising*
Precondition: **Reading behavior is active**
Overseer has approached

1. **Delete reading behavior**
2. Look at Overseer
3. Look at sign
4. Show sudden shock reaction
5. Look at Overseer again
6. Do some quick, sloppy exercises
7. **Spawn** exercise behavior **with high energy**
8. **Add "Watch Overseer" subbehavior to exercise**
9. When Overseer leaves, **gradually reduce energy level of exercise**

FIGURE 5.16: Example of a behavior transition using meta-level controls to combine multiple transition types (meta-level controls are in bold).

agents according to the *signs* and *signifiers* they express instead of the physical actions and behaviors that reflect their internal structure. The agent keeps track of what has been communicated to the user by using the *sign management system*. I surveyed the range of behavior transition types one might want to support, and developed *meta-level controls* to support these transition types by allowing behaviors to refer to one another directly. These controls also allow the designer to express aspects of behavioral interrelationships by making explicit formerly implicit behavioral interactions.

At this point, you should be desperately wondering how these techniques actually worked out in practice. Initial results with them were good or bad, depending on your viewpoint. The transitions clearly reduced the apparent atomization of the agents. Since this was my goal for them, it seemed like I was well on the road to success. However, I did not need to do any fancy user studies to see that straightforward use of transitions per se did *not* improve the comprehensibility of the agent. The agent's behavioral changes were smooth and flowing, but remained just as enigmatic as before.

For example, two of the Patient's low-level signifiers are "watch the Overseer" and "glance around curiously." To change from watching the Overseer to glancing around, I tried using an alternating transition: interleave glances at the Overseer with glances around the junkyard, changing the proportion of glances from each behavior until the Patient was looking only around the junkyard. Clearly, this made the transition between the behaviors smooth; you could not tell when the "looking at

Overseer" behavior ended and the "glancing around" behavior began. On the other hand, you also could not tell *why* the agent was doing that sequence of glances. Watching the Patient, it seemed that its choice of what to look at was pretty arbitrary and not motivated by anything in its environment or, for that matter, in its personality. In fact, it *was* pretty arbitrary, but that was not supposed to be communicated!

In essence, transitions as a form of behavior blending means that the agent changes from randomly jumping between behaviors to randomly morphing between behaviors. While this is certainly less jarring — the user is not constantly notified of random changes by sudden radical changes in agent behavior — it does not fundamentally solve the problem that behavioral choice seems random, not a result of intentional thought. Nevertheless, it seems like transitions such as the guard dog example on page 105 really *should* be able to make the agent's behavioral choices clearer. Where did I go wrong?

For one thing, merely hiding the agent's inadequacies from the user is not enough. The goal for our agents is to be understandable as intentional beings to their audience, for whom these agents should be, according to my own philosophy, explicitly designed. But so far, I have been treating this audience as a bunch of TV-watching couch potatoes who just need to be insulated from the sticky details of agent implementation. That is to say, so far, I have been using transitions merely to hide the agent's atomization from the user, who is seen as a passive observer of the agent's behavior.

In my own defense, I would like to note that I was merely following a grand tradition of post-Eliza AI.[9] Eliza is an extremely simple program intended as a study in natural language communication. It plays the part of a Rogerian psychoanalyst, and basically repeats everything the user says in the form of a question [Weizenbaum, 1965]. To the shock of its programmer and indeed much of the AI community, who knew that Eliza was little more than a language recording and playback device, human users often imputed extraordinary intelligence to Eliza, treating it as a human confidant. The conclusion that many AIers drew from this incident is that human perception of the intelligence of agents is a wildly inaccurate measure of their actual intelligence.

Unfortunately, though, many AI researchers unconsciously go a step further. They conclude that if Eliza's apparent intelligence is a result of a few simple measures, then *any* attempt to be comprehensible to the audience probably merely involves a bunch of 'tricks' that hide the actual stupidity of the agent from the naive and gullible common masses. I must confess shame-facedly that my use of transitions to hide atomization is simply a slightly subtler extension of this attitude. In general, the result of 'Eliza backlash' is research strategies which focus solely on internal or functional aspects of the agent, ones that can be demonstrated to show intelligence without reference to user interpretation. In the end, the user as an active constructor of understanding of the agents is forgotten.

But this minimization of audience involvement is bad for AI because it hinders the development of creatures that truly appear intentional. It turns out, as generations of psychologists, literary critics, and artists

[9]This is another example of the incredible ability of AI Doctrine to hijack my mind despite my explicit anti-Doctrine philosophy

understand, that audiences are not merely passive. They are *actively* constructing understandings of the intentional and pseudo-intentional beings they encounter. Hiding the things that hinder this construction is good; but even better would be *providing tools that support the user in his or her attempt to find meaning in what the agent is doing*. This does not mean cheap tricks that make the agent falsely seem intentional, but support for the user to understand specifically the impression of agenthood (including goals, decision-making processes, thoughts, and feelings) that the designer is trying to get across. The development of architectural mechanisms that support user interpretation will be the technical goal for the rest of the thesis.

In order to support user interpretation, we first need to have a better understanding of how users come to interpret intentional behavior in the first place. We will spend Intermezzo II looking at a case study of how animators use transitions to make their characters come alive. Chapter 6 will look at how the humanities and psychology describe the construction of knowledge about intentional beings. We will use these two sources to figure out how transitions can be used not just to hide the flaws of atomized agents, but to actively support the user's perception of them as intentional beings in the way the designer intends. It will turn out that with a little re-thinking of the nature of transitions, the mechanisms developed in this chapter — signs, signifiers, and sign management; transitions; and meta-level controls — are not only useful for behavior blending, but can also be used to support user interpretation in the way I have described here. I will build on the technical mechanisms introduced in this chapter in the full development and evaluation of the Expressivator in Chapter 7.

Intermezzo II

Luxo, Jr.: A Case Study of Transitions in Animation

In Chapter 5, I pushed the technical understanding of schizophrenia as far as it would go. The result was some interesting technology that helped to reduce schizophrenia, while miraculously avoiding making agents seem more intentional. You should note that this handily solves the technical problem, but manages to do it while ignoring or even subverting the big picture that motivated the technical problem.[1]

Let's take a moment to go back to the basics. The dream is to be able to create artificial creatures, whether built as robots or rendered by computer graphics, that are not merely smart but really seem alive and intentional. These agents would come to life like characters in a novel or film, that, although human creations, seem to have a life of their own. Although we know they are in some sense 'fabrications,' we listen to them, sympathize with them, laugh at them, hate them, fall in love with them, without a sense of being deluded. Their concerns, worries, and life dilemmas are not simply factual; they are at times ridiculous, at times meaningful, but always to be interpreted within the full context of human life.

What would such artificial creatures look like? One way of finding out is to do a thought experiment. We already know that such creatures can be generated, not by an AI program, but by a character animator. What if we pretend that this animation is actually the result of a behavior-based AI program? Could we reverse-engineer the program that generated it?

The idea that character animators have something to tell computer scientists about how to build agents is not novel. This idea has already been explored by several AI researchers starting with Joseph Bates [Bates, 1994]. In this Intermezzo, I will add to this tradition by looking at a particular animated sequence as though it were generated by an AI program, and then imagine how behaviors and transitions were used to create the feeling that the character is really intentional. How are the different 'behaviors' of the 'agents' connected? How do these connections help to make the agent alive?

Clearly, it is unlikely that animators actually think in terms of 'behaviors' and 'transitions,' as an AI researcher would. Nevertheless, we can

[1] This is perhaps a larger (though unwanted) tradition in science.

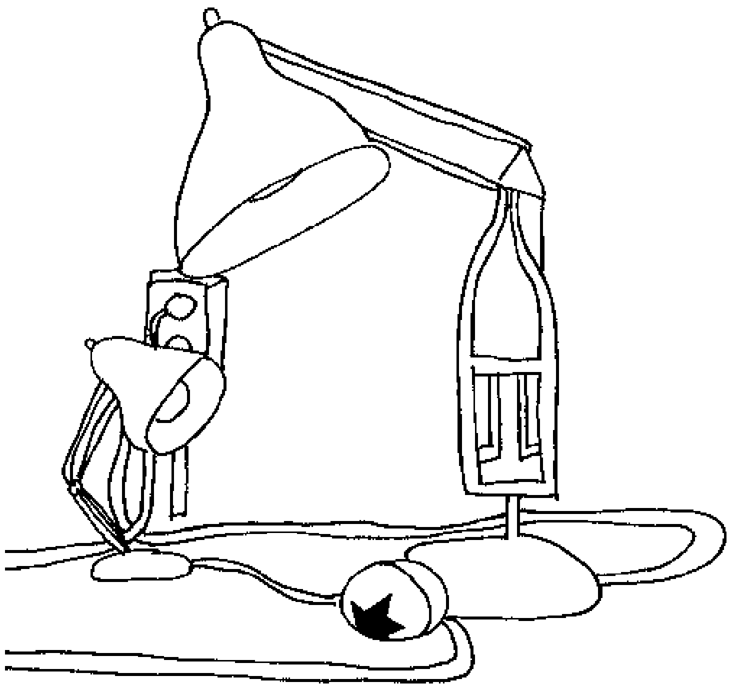


FIGURE II.1: Luxos Senior and Junior (artist's rendition)

learn something by provisionally viewing animation through the lens of AI architecturology. It turns out that animation brings an interesting, new perspective to the table, in the ways that it both is and is not adequately described by the behavioral metaphor.

Introduction to *Luxo, Jr.*

"Whether it is generated by hand or by computer, the first goal of the animator is to entertain. The animator must have two things: a clear concept of what will entertain the audience; and the tools and skills to put those ideas across clearly and unambiguously. Tools, in the sense of hardware and software, are simply not enough. The principles discussed in this paper, so useful in producing 50 years of rich entertainment, are tools as well... tools which are just as important as the computers we work with." ([Lasseter, 1987], 43)

The animation we will be looking at is John Lasseter's short film *Luxo, Jr.*[Pixar, 1986], an artist's rendition of which appears in Figure II.1.[2] This film was one of the first computer animations to focus on developing character and intentionality, rather than on creating mechanical photorealism. Lasseter's explicit goal is to use traditional (hand-drawn) animation techniques to communicate personality, emotion, and intentionality clearly through his computer-generated images [Lasseter, 1987]. The success of *Luxo* and subsequent films such as *Toy Story* suggests that he has been effective.

The movie itself centers on two characters, Luxos Junior and Senior, and a ball. Luxo Junior comes on stage, playing with the ball. After some time, the ball breaks. Luxo Junior is at first disappointed, but soon finds a new ball. Despite (or perhaps because of) the utter simplicity of the plot, the characters are strongly portrayed, clearly emotional and intentional, and fun to watch.

[2]Permission to use an actual still from the film was not given.

You may note a striking family resemblance between the Luxos and the Patient in the Industrial Graveyard.[3] More importantly, the relatively simple structure of the lamps in *Luxo*, the simplicity of its plot and the agents' behavior, the absence of natural language, and the fact that it is all rendered by computer mean that, perhaps, the goal of automatically generating similarly affective characters is not entirely implausible, though perhaps far beyond the state of the art. Let's imagine that they *are* created by a behavior- and transition-based architecture. What can this tell us about how transitions work?

Luxos As AI Agents

Detailed analysis of behaviors and transitions in *Luxo* can be found in Appendix B. The general trend is that *agents communicate what they will do before they do it*. This means they stop whatever they are doing and engage in some pre-behavior activity to tell you what they are going to do next. This use of transition corresponds to the animation technique of anticipation.

> Anticipation is... a device to catch the audience's eye, to prepare them for the next movement and lead them to expect it before it actually occurs. Anticipation is often used to explain what the following action is going to be. Before a character reaches to grab an object, he first raises his arms as he stares at the article, broadcasting the fact that he is going to do something with that particular object. The anticipitary moves may not show *why* he is doing something, but there is no question about *what* he is going to do next. ([Lasseter, 1987], 38)

This is different from the default transition theory of Chapter 5. There, we used transitions to blend together two behaviors. In this mindset, the important thing is to finish up the old behavior cleanly and begin the new behavior in an unobtrusive way. But with Luxos the old behavior is at least somewhat irrelevant. The point of transitions here is that the character must do some communication before it starts a behavior. This communication tells the audience that the Luxo has made a decision to do something different, as well as letting the audience know how the behaviors interrelate.

Transitions communicate a variety of such relationships in *Luxo, Jr.*:

- That didn't work; I have a new idea.
- Hey, what just happened?
- Oh no! Let me get out of here!
- I wonder what *that* will do?

These various relationships are largely communicated through a small set of basic tools.

[3] The Patient is for this reason sometimes nicknamed "Lixo," or, in moments of hacking frustration, "Suxo."

- *Eye movement* — This is probably the single most important way Luxos communicate behavioral transition. They stop, look at what they are going to do, and then do it. The moment of looking is important as it communicates that the lamp is making a decision.

  Just watching by itself can also be a behavioral transition. As a default, if the character does not know what to do, it can just watch what is going on. When something new has caught the character's attention, it can change to a behavior involving that object.

- *Behavioral blend* — Where behaviors correspond to movements, the two behaviors can be blended using low-level action blending techniques like those presented in the previous chapter. For example, when Luxo moves from standing still to examining an item, it starts out very slowly (almost still), then gradually speeds up. When Luxo moves from sighing to hopping, it does sad, sighing hops. In these cases, the animator has found a defining characteristic of one behavior, and blends the behaviors by applying that characteristic to the other behavior.

  Again, what is important here is not to blend the behaviors per se but the impression of that behavior on the user. If some behaviors can be fundamentally defined rather simply, it will be easy to mix them in with other ones. You are not including the whole behavior, but an *image* of it.

- *Alternation* — At times, Luxo transitions between behaviors by switching between parts of them. For example, when Luxo Senior switches from watching the ball to watching Luxo Junior, it alternates glances between them.

- *Shock reaction* — a common transition. The agent engages in some behavior, then shows a shock reaction to something in the environment and switches to a different behavior. This shows clearly that the agent is reacting to something unexpected rather than just changing on a whim.

- *Shared object* — Often the old and new behaviors share an object of interest. Transitions are frequently predicated on external objects upon which the character focuses during a transition. This makes the transition clearly not internal or arbitrary, but a reaction to observable events.

- *Off-screen* and/or *non-individualistic* — At times Luxo will switch behaviors off-screen. Here the change in behavior will be reflected in the reaction of the character left on the screen. This means not all behavioral transitions take place in the creature him/herself — some transitions are communicated by the reaction of the other character. This in turn implies that transitions are not just about individual behavior, but (at least in Luxo) are important in terms of story — they are about advancing the story, and can therefore appear in either character. Additional support for this is in the fact that Junior and Senior generally do not transition at the same time — while we are watching Junior play Senior just stays in one (simple background) behavior.

In general, unlike the transitions of Chapter 5, in Luxo most transitions are not 'internal' or 'arbitrary.' They are *reactions* to observable events: a result of a previous action or something another creature did. Transitions relate the events of the story to one another by expressing the relationships between behaviors and explaining why the creature is moving from one activity to the next. They help the audience to understand what Luxo is doing by anticipating and explaining the reasons for the behavior in which it engages.

Luxos Exceed AI

While viewing *Luxo* through the lens of behaviors and transitions is illuminating, there are clearly some ways in which this paradigm does not do justice to the film. These areas point to some fundamental limitations in the behavior/transition metaphor. These limitations are not all addressed in this thesis, but are mentioned here to provide a roadmap to changes which may need to be made to generate truly expressive agents.

Inadequacies of Behaviorism

The first step in analyzing Luxo's transitions is to identify the behaviors and transitions Luxo uses. But a number of behaviors cannot be classified easily as 'behavior' or 'transition'. The most obvious one is 'watch,' in which Senior engages for much of the film. 'Watch' is a transition because it fills in spaces between activities, telling you what Senior is thinking about and deciding to do. It is also a behavior because it is so long, and because it really seems to be an activity in and of itself.

In addition, some 'behaviors' seem to exist only in a transitory phase. A good example of this is when Junior hops on stage for the first time, playing with the ball. It then spends some time alternating between looking at Senior and looking at the ball (the 'transition'), and hops off stage to go play with the ball again (the 'original behavior').

More fundamentally, while you could provisionally call much Luxo activity "behaviors," Luxo's behaviors are clearly different from behavior in the behavior-based sense. For example, Luxo's behaviors are not repeatable; when he engages in the 'same' behavior twice it is often quite different in its presentation and context. It seems inaccurate to call them "really" or "fundamentally" the same thing.

In general, AI-style behaviors carry with them a load of intellectual baggage that animators do not seem to want.

- For AI researchers, a behavior fundamentally *is* the name or concept of the behavior. For animators, behavior is movement that may or may not be described with a particular name; while this name may repeat ("doing the same behavior twice"), the action itself may not.

- For AI researchers, an agent moves from behavior to behavior, and is always running at least one. For animators, an agent is always engaging in movement, which is interpretable as an activity, or shows the agent's emotions, or reveals the agent's motivations, or...

Fundamentally, for AI researchers, the behavioral level is reality, with the actions a surface impression of this deeper level. For animators, the actions are reality, with the behaviors an abstract description of the real. AI thought seeks out the deep structure of agent action and finds it in behaviors; animation thought seeks out the clear communication of ideas and finds it in all the details of character movement.

Inadequacies of Transitionism

If behaviors are not completely adequate to understanding Luxo's activity, then it should come as little surprise that transitions are not, either. The non-individualistic transitions mentioned on page 136 are one interesting way in which the idea of behavior transitions needs to be 'bent' in order to fit Luxo behavior. Behavior transitions as conceptualized so far have resided purely in one individual, i.e. one behavior changing to another in a clear way. Non-individualistic transitions expand this notion for when a group of agents is meant to have a cumulative effect, rather than focusing on each individual agent.

Non-individualistic transitions, by exceeding the definition of transitions made in the last chapter, reveal an inadequate assumption underlying this definition. This assumption, which comes from the behavior-based AI tradition, is that all behavior is somehow fundamental to the individual, rather than to the group to which the individual belongs. In this tradition, even multi-agent systems that engage in group coordination tend to work by figuring out how to program the individual agents so that the correct global behavior emerges from local interactions based on local knowledge. In contrast, for animation, the *story* is fundamental; the characters are secondary. The decision of which behavior a character should present is not based primarily on its plausibility for that character but on how it fits into the overall plot.

This suggest that animation has a fundamentally different understanding of the relationships between the parts and the whole. In AI, the 'parts' (agents) are primary, with the whole being the simple sum of the parts. This corresponds exactly with the whole agent being the simple sum of the individual behaviors. In animation, on the other hand, the 'whole' is primary, with the 'parts' (characters) being instantiations of and motivated for the whole. This different way of conceptualizing the relationship between part and whole is a fundamental difference between humanistic and scientific worldviews. It will become key in Chapter 6.

Transitions from an Animator's Perspective

These differences between the AI and animation worldviews suggest that someone trained in animation may come to quite different conclusions about how the idea of transitions applies to *Luxo, Jr.*. To fill out this analysis, I asked a professional animator, Steve Curcuru, to do an informal analysis of Luxo in parallel with mine [Curcuru,]. Since Curcuru had at that point not yet become infected with any knowledge of behavior-based AI, his impressions are based on how an animator might think of behaviors and transitions, and are therefore, unsurprisingly, quite different from mine.

In general, Curcuru focuses much more on the actual form and structure of Luxo's motion, whereas I — with my AI intellectual baggage firmly in tow — tend to focus on what the character is 'fundamentally' doing. Curcuru therefore, unlike me, tends to find transitions within the details of Luxo's movements. For example, Curcuru points out that a number of times, Luxo 'settles' from one behavior to another. That is, if a new behavior is relatively static in terms of motion (e.g. Senior looking off-screen), the character will slowly move from the position of the old behavior into its new final position. Also, Curcuru describes how a quick motion *contrasts* with a slow motion the character just engaged in, and that this contrast is essential to understanding what the character was doing (a change in thought). This suggests that the relationship between the agent's behaviors may be more complicated that a simple transition that can be inserted between them; in these cases it is a relation between the *forms* of the two behaviors.

Curcuru additionally makes clear that the idea that behavior is intended to communicate permeates not only transitions but also behaviors themselves. He identifies many aspects of Luxo's behavior that are there simply to show what the agent is thinking. He mentions two major tools in particular for showing what a character is thinking: anticipations and holds. Anticipations may be helpful to get the audience to understand what the character is doing and / or to make the agent seem more intentional. Holds are used to depict that the character is thinking.

Curcuru believes that transitions are fundamentally there to show *why* the character's behavior changes. He describes Luxo's transitions as having the following general form:

1. The character does something.

2. Hold; the character must be thinking about something.

3. The character does something different; hence, it must have changed its mind.

During the transition, the character shows that it is considering something, usually an observable object or event. When the behavior changes, the audience assumes that the change is due to this moment of thought.

The fundamental insight from Curcuru's analysis is that *transitions show that the character is making behavioral changes reflectively, rather than reflexively*. The character is not instinctively or arbitrarily moving from action to action, but is considering what it does. Transitions allow the animator to make clear that the character is noticing the world around it and reacting to it in its own idiosyncratic ways. Done well, this thoughtful interactivity makes the character come alive.

Lessons Learned from Luxo

This analysis of Luxo leads to some new conclusions about transitions. In Chapter 5, transitions were intended to make sure that behavior change is not abrupt. The idea was that, if abrupt and sudden behavior switching is confusing to the user, then we should disguise the behavioral change so the user does not notice it.

But this analysis of Luxo shows that transitions do not merely blend together behaviors in a seamless whole. In Luxo, transitions are needed, not to hide behavioral change, but to set the stage for *new* behavior. They prepare the audience for the new behavior by anticipating it and by showing the reasons for the switch. They let the agent unmistakably show that its behavior is affected by what is happening around it.

For Luxo, transitions are about the *reasons* for behavior change. They show why the agent is moving from activity to activity. Transitions show that the agent is making its behavioral choices reflectively, not instinctually, by revealing the agent's thinking processes. They are therefore essential to giving the agent the aura of being a conscious being, rather than an automaton.

Luxo shows that transitions are intended to communicate; and *so are behaviors*. Many aspects of Luxo's behaviors are there purely to show what Luxo is thinking. Therefore, design choices that let transitions communicate better may also be useful for improving regular behaviors. Disciplined use of transitions and the architectural mechanisms that support them may help make *all* behavior clearer, not just the behaviors that are directly related with transitions.

Chapter 6

Narrative Intelligence

At this point, let's take a moment to review where we are. In Chapter 2, we started with the problem of incoherence and mechanicity in autonomous agents. As agent builders combine more and more behavior, the overall activity of the agents tends to degrade into a jumping-around between separately defined behaviors, a problem we have been calling *schizophrenia*.

In Chapters 3 and 4, we looked at schizophrenia and AI in culture, suggesting that one way of addressing the problems of schizophrenia is by looking at agents in their social and cultural context. This motivated us to redefine the problem of schizophrenia in Chapter 5 in terms of the user's perception. Instead of asking "how can an agent be coherent?" we ask, "how can an agent appear coherent to the user?"

This reformulation suggests that we should use transitions to smooth between behaviors, i.e. to *hide* the breaks between behaviors from the user. However, as we noted at the end of Chapter 5, transitions as de-atomization do not really address the fundamental problem of schizophrenia. They do hide the breaks between behaviors, but they do not do so in a way that makes the agent seem any more intentional.

In Intermezzo II, we saw that character animators have a fundamentally different way of thinking about transitions between behaviors. Instead of using transitions to hide or smooth over a behavioral change, transitions are used to help the user understand the *reasons* for behavioral change. Far from hiding behavioral switches, transitions call attention to them, but they do so in such a way that they help the viewer to figure out what the agent is doing. Transitions are one tool among many that animators use in order to send cues to the viewer about how they should interpret the character.

This animation viewpoint suggests that we have been looking at the problem of agent construction from the wrong end. Rather than focusing on the agent — "how can we fix the agent so that the user will not notice it is actually incoherent?" — we should focus on the *user*. This leads to yet another re-formulation of the problem statement: "*how can we support the user in constructing coherent interpretations of the agent?*"

This will be the final reformulation of the problem statement, leading to the full-blown Expressivator architecture of Chapter 7. In this chapter, we will explore the ramifications of thinking about the problem in this way. First, we will review how this problem statement diverges from traditional AI thinking. Then, we will find out how people interpret intentional behavior through a foray into narrative psychology. Finally, I will present principles of *Narrative Intelligence*, i.e. a way of designing agents to support the user's construction of narrative interpretations of their behavior.

Interpretation and Agent Transparency

Many AI systems that try to make agent behavior clear are based on what I call the Agent Transparency Assumption. In AI, the agent is not thought of as the viewer's interpretation, but exists 'objectively', i.e. is an independent object to be observed passively. The agent is thought to primarily exist on its own, with its presentation an afterthought. Therefore, in AI communication of the 'idea' of the agent is thought to be best achieved, not by tailoring the visible presentation of the agent towards particular interpretations, but by allowing the user to see what the agent is 'actually' doing. Here 'actual' means 'in the agent's code' — i.e., the way the designer thinks of the agent. Blumberg, for example, defines the expression of intentionality as "allow[ing] the observer to understand on which goal the system is working, how it views its own progress towards that goal, and what the system is likely to do next" ([Blumberg, 1996], 25). For AI, the character actually, independently exists — as represented in the body of its code — and the interpretation of the viewer is not a creative act but a passive observation or correctable reconstruction of the agent's code.

But even in AI, users are involved in a complex process of interpretation of the agent's behavior. This is because the user's view of the agent is quite different from the designer's. Agent designers tend to think of agents in terms of the code we use to write them. We choose particular goals, emotions, or plans for the agent, and when we watch the agent, we interpret its activity according to those components. We are on the lookout for the behaviors and emotions we know it must have, since we put them in the code.

However, users who do not know how the agent was designed do not have the internal structure of the agent as a resource in interpreting the agent's activity. *All the user can go on is the agent's physical actions*. The agent's "actual" structure (goals, behaviors, and so on) must be inferred from the movements the user sees the agent use.

Given this relative poverty of information, *it is amazing users understand agents at all*! It is fairly incredible that, when users observe two spheres with eyeballs flattening and moving towards each other, they quite frequently say "hey! look! They're getting into a fight!" Extremely simple physical cues often lead users to infer complex motives and behavior that may or may not be warranted by the code running the agent.

Viewers' understanding of agents is grounded in the fact that people are fundamentally social creatures, specialized in understanding intentional behavior.[1] When people watch our agents, they bring with them

[1] The degree to which this is true can be understood by looking at the great handicaps

sophisticated capacities of interpretation which often allow them to infer or "read in" the intentional behavior we would like them to see in our creatures — even despite the obstacles we agent builders often put in their way! Even when people happen to reconstruct understandings of our agents that correspond to what we designed into their code, they are in no sense passively observing; it is always an active reconstruction.

In AI, we generally feel that this process of interpretation is somewhat dubious. Instead of encouraging the user to interpret to his or her heart's content, we try to ground the user's interpretation of the agent in the 'actual' agent, i.e. its code. That is, we try *to make the user look at the agent in the same way the designer does*. The major problem with this strategy is that it is counterproductive. On the one hand, users are used to interpreting creatures' behavior, and they will resist attempts to 'see' in ways that are different from what they are used to. On the other, users are extremely good at interpreting creatures' behavior, so we are wasting their talents.

Animation suggests a different strategy: maybe we should *try to make the designer look at the agent in the same way the user does*. The animation viewpoint suggests that rather than throwing out this interpretive ability by getting users to simply 'see' the code, we can make our creatures appear maximally intentional by *supporting users in their ongoing drive to interpret the agent as an intentional creature*. That is, we can construct agents so that they give off cues that are easy for users to understand as coherent, intentional behavior.

One way of understanding this reformulation is to go back to the very concept of agent. As mentioned in Chapter 1, we use the notion of 'agent' when we think that it is helpful, informative, or good P.R. to think of our programs as self-contained individuals. The term agent has become mind-numbingly popular recently and has been substantially diluted in the past several years, so that now people use the word 'agent' almost interchangeably with the word 'program' or 'engineered artefact' (as in, "I used my remote control agent to turn on my TV agent"). Autonomous agent researchers such as myself have felt alternately encroached upon and far superior to the competitition, since our usage of the term 'agent' — to refer to a computer-controlled character or artificial creature roughly analogous to living agents — seems to be one of the few actually meaningful uses of the term.

Here I would like to suggest that, despite the moral high ground autonomous agent researchers occupy in this respect, *the usage of the agent metaphor for autonomous agents may actually be unhelpful*. As discussed in Chapter 4, thinking about our programs as 'agents' implies that they are autonomous and self-contained, and that communication of agent activity to users consists of the apperception by external people of an independently and objectively existing object. Animation and narrative psychology suggests that for applications where human comprehension of our agents is essential, it may be more helpful to think of autonomous agents *as narrative*. This implies that an agent is not self-contained, but exists through a process of communication. An agent-as-narrative has an author and an audience, exists in a context that affects how it is understood, and comes to life only in so far as it is adequately communicated

that autistic people face in our society.

to the audience.

In this chapter, we will explore what it means for agents to be structured and communicated as narrative. In the next section, we will start by looking at narrative psychology, which studies how people interpret specifically intentional behavior. Narrative psychology suggests that this process of creating narrative is the fundamental difference between the way people understand intentional beings and mechanical artefacts. This implies that by structuring our agents as narrative, we can make it more natural for people to understand our agents as comprehensible, intentional beings. I will therefore discuss how agents can be built according to the principles of narrative. This forms a style of agent-building I term *Narrative Intelligence*, in which agents give off visual behavioral cues that are easy to assimilate into narrative.

Principles of Narrative Psychology
or How We (Sometimes) Make Sense of Creatures

Artificial Intelligence attempts to generate intentional creatures by blurring the distinction between biological, living beings and automatic processes of the kind that can be run on computers. That is, AI agents should ideally be understandable both as well-specified physical objects and as sentient creatures. But it turns out that human understanding of the behavior of humans and other conscious beings differs in important ways from the way we understand the behavior of such physical objects as toasters. Identifying the distinction between these two styles of comprehension is essential for discovering how to build creatures that are understandable not just as helpful tools but as living beings.

The way people understand meaningful human activity is the subject of narrative psychology, an area of study developed by Jerome Bruner [Bruner, 1986] [Bruner, 1990]. Narrative psychology shows that, whereas people tend to understand inanimate objects in terms of cause-effect rules and by using logical reasoning, intentional behavior is made comprehensible by structuring it into narrative or 'stories.' We find structure, not by simply observing it in the person's activity, but through a sophisticated process of interpretation. This interpretation involves finding relations between what the person does from moment to moment, speculating about what the person thinks and feels about his or her activity, and understanding how the person's behavior relates to his or her physical, social, and behavioral context.

Even non-experts can effortlessly create sophisticated interpretations of minimal behavioral and verbal cues. In fact, such interpretation is so natural to us that when the cues to create narrative are missing, people spend substantial time and effort trying to come up with possible explanations. This process can be seen in action when users try to understand our currently relatively incomprehensible agents!

This sometimes breathtaking ability — and compulsion — of the user to understand behavior by constructing narrative may provide the key to building agents that truly appear alive. *If humans understand intentional behavior by organizing it into narrative, then our agents will be more 'intentionally comprehensible' if they provide narrative cues.* That is to say, rather than simply presenting intelligent actions, agents should give visible cues that support users in their ongoing mission to generate narrative explanation of an agent's activity. We can do this by organizing our agents so that their behavior provides the visible markers of narrative. The remainder of this chapter presents the properties of narrative and explains how they can be applied to agent construction.

Prolegomena to a Future Narrative Intelligence

There has recently been a groundswell of interest in narrative in AI and human-computer interaction (HCI). Narrative techniques have been used for applications from automatic camera control for interactive fiction [Galyean, 1995] to story generation [Elliott *et al.*, 1998]. Abbe Don and Brenda Laurel argue that, since humans organize and understand their experiences in terms of narrative, computer interfaces should be organized as narrative, too [Don, 1990] [Laurel, 1991] [Laurel, 1986]. Similarly, Kerstin Dautenhahn and Chrystopher Nehaniv argue that robots may be able to use narrative in the form of autobiography to understand both themselves and each other [Dautenhahn and Nehaniv, 1998].

The term Narrative Intelligence has been used by an informal group at the MIT Media Lab to describe this conjunction of narrative and Artificial Intelligence. It is also used by David Blair and Tom Meyer to refer to the human ability to organize information into narrative [Blair and Meyer, 1997]. Here, I want to suggest that Narrative Intelligence can be understood as the confluence of these two uses: *that artificial agents can be designed to produce narratively comprehensible behavior by structuring their visible activity in ways that make it easy for humans to create narrative explanations of them.*

In order to do this, we need to have a clear understanding of how narrative works. Fortunately, the properties of narrative have been extensively studied by humanists. Bruner (nonexhaustively) lists the following properties [Bruner, 1991]:

- *Narrative Diachronicity*: Narratives do not focus on events moment-by-moment, but on how they relate over time.
- *Particularity*: Narratives are about particular individuals and particular events.
- *Intentional State Entailment*: When people are acting in a narrative, the important part is not what the people do, but how they think and feel about what they do.
- *Hermeneutic Composability*: Just as a narrative comes to life from the actions of which it is composed, those actions are understood with respect to how they fit into the narrative as a whole. Neither can be understood completely without the other. Hence, understanding narrative requires interpretation in a gradual and dialectical process of understanding.
- *Canonicity and Breach*: Narrative gets its 'point' when expectations are breached. There is a tension in narrative between what we expect to happen, and what actually happens.

- *Genericness*: Narratives are understood with respect to genre expectations, which we pick up from our culture.
- *Referentiality*: Narratives are not about finding the absolute truth of a situation; they are about putting events into an order that feels right.
- *Normativeness*: Narratives depend strongly on the audience's conventional expectations about plot and behavior.
- *Context Sensitivity and Negotiability*: Narrative is not 'in' the thing being understood; it is generated through a complex negotiation between reader and text.
- *Narrative Accrual*: Multiple narratives combine to form, not one coherent story, but a tradition or culture.

While these properties are not meant to be the final story on narrative, they stake out the narrative landscape. Taking narrative agents seriously means understanding how these properties can influence agent design. It will turn out that current AI techniques, which largely inherit their methodology from the sciences and engineering, often undermine or contradict the more humanist properties of narrative. Here, I will explain problems with current agent-building techniques, techniques already in use that are more amenable to narrative, and potential practices that could be more friendly to the goal of meaningful Narrative Intelligence. This will form the theory or philosophy of Narrative Intelligence; its technical manifestation will rear its head in the next chapter.

One note of caution: the goal here is to interpret the properties of narrative with respect to agent-building. This interpretation is itself narrative. Since, as we will see below, the nature of narrative truth is different from that of scientific factuality, this essay should not be read in the typically scientific sense of stating the absolute truth about how narrative informs AI. Rather, I will look at the properties of narrative in the context of current AI research, looking for insights that might help us to understand better what we are doing better and suggest (rather than insist on) new directions.

1.Narrative Diachronicity

The most basic property of narrative is its diachronicity: a narrative relates events *over time*. Events are not understood in terms of their moment-by-moment significance, but in terms of how they relate to one another as events unfold. For example, if Fred has an argument and then kicks the cat, we tend to infer that the cat-kicking is a result of his frustration at the argument. When people observe agents, they do not just care about what the agent is doing; they want to understand the relations between the agent's actions at various points in time. These perceived relations play an important role in how an agent's subsequent actions are understood. This means that, to be properly understood, it is important for agents to express their actions so that the intended relationships are clear.

However, as described in Chapter 2, it is currently fashionable to design behavior-based autonomous agents using action-selection, an agent-building technique that ignores the diachronic structure of behavior.

Action-selection algorithms work by continuously redeciding the best action the agent can take in order to fulfill its goals [Maes, 1989a]. Because action-selection involves constantly redeciding the agent's actions based on what is currently optimal, there is no common thread structuring the actions that are chosen into understandable sequences — this fact is simply schizophrenia rephrased. Schizophrenia undermines the appearance of intentionality because agent action seems to be organized arbitrarily over time, or, at maximum, in terms of automatic stimulus-response.[2]

More generally, as mentioned in Chapter 5, expressing the relationships between behaviors is not well-supported in most behavior-based systems (a complaint also raised in [Neal Reilly, 1996]). While these architectures do provide support for clear, expressive *individual* behaviors, they have problems when it comes to expressing relations *between* behaviors. This is because a typical behavior-based system (e.g. [Blumberg, 1994] [Brooks, 1986a] [Maes, 1989b]) treats each behavior separately; behaviors should refer as little as possible to other behaviors. Because of this design choice, a behavior, when turned on, does not know why it is turned on, who was turned on before it, or even who else is on at the same time. It knows only that its preconditions must have been met, but it does not know what other behaviors are possible and why it was chosen instead of them. In most behavior-based architectures, behaviors simply do not know enough about other behaviors to be able to express their interrelationships to the user.

In this light, classical AI would seem to have an advantage over alternative AI, since it is explicitly interested in generating structured behavior through such mechanisms as scripts and hierarchical plans. However, classical AI runs into similar trouble with *its* modular boundaries, which occur not between behaviors but between the agent's functionalities; for example, the agent may say a word it cannot understand. Fundamentally, agent-building techniques from Marvin Minsky's Society of Mind [Minsky, 1988] to standard behavior-based agent-building [Maes, 1991] to the decomposition of classical agents into, for example, a planner, a natural language system, and perception [Vere and Bickmore, 1990] are all based on divide-and-conquer approaches to agenthood. Being good computer scientists, one of the goals of AI researchers is to come up with modular solutions that can be easily engineered. While some amount of atomization is necessary to build an engineered system, narrative intentionality is undermined when the parts of the agent are designed so separately that they are visibly disjoint in the behavior of the agent. Schizophrenia is an example of this problem, since when behaviors are designed separately the agent's overall activity is reduced to a seemingly pointless jumping around between behaviors. Bryan Loyall similarly points out that visi-

[2]This is unfortunate, since the original idea of constantly redeciding behavior came in work explicitly interested in diachronic structure. Philip Agre and David Chapman focus, not on the design of the agent per se, but on the ongoing dynamics of the agent and the environment [Agre and Chapman, 1987]. The goal is to construct action-selection so that, when put in a particular environment, agents *will* tend to have particular diachronic structure in their behavior. Continuous redecision is part of this work because it keeps the agent's actions closely tied to the agent's context, a property that is also important for narrative, as we will see below. However, the concept of the action-selection algorithm itself tends to undermine diachronic structure, especially when it is used for agent — rather than dynamic — design.

ble module boundaries destroy the appearance of aliveness in believable agents [Loyall, 1997a].

The end result is that the seductive goal of the plug-n-play agent — built from the simple composition of arbitrary parts — may be deeply incompatible with intentionality. Architectures like that of Steels [Steels, 1994], which design behaviors in a deeply intertwined way, make the agent design process more difficult, but may have a better shot at generating the complexity and nonmodularity of organic behavior. In Chapter 7, we will try a less drastic solution, using transition sequences to relate separately designed behaviors.

2.Particularity

Narratives are not simply abstract events; they are always particular. "Boy-meets-girl, boy-loses-girl" is not a narrative; it is the structure for a narrative, which must always involve a particular boy, a particular girl, a particular way of meeting, a particular way of losing. These details bring the story to life. However, details do not by themselves make a narrative either; the 'abstract structure' the details can be ordered into brings meaning to the details themselves. A narrative must be understood in terms of tension between the particular details and the abstract categories they refer to; without either of these, it is meaningless.

This same tension between the abstract and the particular can be found in agent architectures. Agent designers tend to think about what the agent is doing in terms of abstract categories: the agent is eating, hunting, sleeping, etc. However, users who are interacting with the agent do not see the abstract categories; they only see the physical movements in which the agent is engaged. The challenge for the designer is to make the agent so that the user can (1) recognize the particular details of the agent's actions and (2) generalize to the abstract categories of behavior, goal, or emotion that motivated those details. Only with a full understanding at both the particular and the abstract levels will the user be likely to see the creature as the living being the designer is trying to create.

But AI researchers are hampered in this full elucidation of the dialectical relationship between the particular and the abstract by the valorization of the abstract in computer science. As mentioned in Intermezzo II, in AI we tend to think of the agent's behaviors or plans as what the agent is 'really' doing, with the particular details of movement being a pesky detail to be worked out later. In fact, most designers of agents do not concern themselves with the actual working out of the details of movement or action at all. Instead, they stop at the abstract level of behavior selection, reducing the full complexity of physical behavior to an enumeration of behavior names. Maes, for example, uses abstract atomic actions such as "pick-up-sander" [Maes, 1989b].

Similarly, the Oz Project's first major virtual creature, Lyotard, was a text-based virtual cat [Bates *et al.*, 1992]. Because Lyotard lived in a text environment, his behaviors were also text and therefore high level: "Lyotard jumps in your lap," "Lyotard eats a sardine," "Lyotard bites you." Because we were using text, action did not need to be specified at a more detailed level. We did not have to specify, for example, how Lyotard moved his legs in order to jump in your lap.

Lyotard's successors, the Woggles, on the other hand, were graphically represented. As a consequence, we were forced into specifically defining every low-level action an agent took as part of a behavior. The effort that specification took meant that we spent less time on the Woggles' brains, and as a consequence the Woggles are not as smart as Lyotard. But — surprisingly to us — the Woggles also have much greater affective power than Lyotard. People find the Woggles simply more convincingly alive than the text cat, despite the fact that Lyotard is superior from an AI point of view. This is probably in part because we were forced to define a particular body, particular movements, and all those pesky particularities we AI researchers would rather avoid. [3]

Again, as mentioned in Intermezzo II, if we look at animation, the valorization tends to run to the other extreme [Thomas and Johnston, 1981]: the particular is the most essential. Animators tend to think mostly at the level of surface movement; this movement may be interpretable as a behavior, as evidence of the character's emotions, as revealing the character's motivations, or as any of a host of things or nothing at all. Animators make the point that any character is of necessity deeply particular, including all the details of movement, the structure of the body, and quirks of behavior. The abstract comes as an afterthought. Certainly, animators make use of a background idea of plot, emotion, and abstract ideas of 'what the character is doing,' but this is not the level at which most of animators' thinking takes place.

Loyall points out that this focus on the particular is also essential to the creation of effective believable agents [Loyall, 1997a]. A focus on particularity by itself, though, is not adequate for creating artificial agents. Agents are expected to interact autonomously with the user over time. In order to build such autonomous systems, we need to have some idea of how to structure the agent so that it can recognize situations and react appropriately. Because we do not know every detail of what will happen to the agent, this structure necessarily involves abstract concepts in such aspects as the modules of the agent, the classification of situations according to appropriate responses, abstract behaviors, emotions, goals, and so on.[4] We must design agents, at least partially, at an abstract level.

In order to build agents that effectively communicate through narrative, AI researchers will need to balance their ability to think at the abstract level with a new-found interest in the particular details their system produces, an approach that seems to be gaining in popularity [Frank *et al.*, 1997]. Narrative Intelligence is only possible with a deep-felt respect for the complex relationship between the abstract categories that structure an agent and the physical details that allow those categories to be embodied, to be 'read,' and to become meaningful to the user.

3.Intentional State Entailment

Suppose you hear the following:

> A man sees the light is out. He kills himself.

[3] Similar arguments may hold for robots. The Sony robotic dogs at Agents '97 were a compelling demonstration that robots may have much greater affective power than even graphically represented agents [Fujita and Kageyama, 1997].

[4] It may be that one day we can use machine learning to develop this structure instead; whether this learned agent must also be structured abstractly remains to be seen.

Is this a story? Not yet. You don't understand it. After endless questions, you find out that the man was responsible for a light house. During the night, a ship ran aground off shore. When the man sees that the light house light is out, he realizes that he is responsible for the shipwreck. Feeling horribly guilty, he sees no choice but to kill himself. Now that we know what the man was thinking, we have a story.

In a narrative, what 'actually happens' matters less than what the actors feel or think about what has happened. Fundamentally, people want to know not just what happened but *why* it happened. This does not mean the 'causes' of an event in terms of physical laws or stimulus-response reactions, but the reasons an actor freely chose to do what s/he did. The narrative is made sense of with respect to the thoughts and feelings of the people involved in its events.

This means that when people watch autonomous agents, they are not just interested in what the agent does. They want to know how the agent thinks and feels about the world around it. Instead of just knowing what the agent has chosen to do, they want to know *why* the agent has chosen to do it. This is, in fact, the grounds for the strategy animation uses for transitions: as mentioned in Intermezzo II, transitions in animation communicate the reasons for behavioral change.

But in many autonomous agent architectures, the reasons for the decisions the agent makes are part of the implicit architecture of the agent and therefore not directly expressible to the user. Bruce Blumberg's Hamsterdam architecture, for example, represents the appropriateness of each currently possible behavior as a number; at every time step the behavior with the highest number is chosen [Blumberg, 1996]. With this system, the reasons for behavioral choice are reduced to selecting the highest number; the 'actual' reason that behavior is the best is implicit in the set of equations used to calculate the number. The agent simply does not have access to the information necessary to express why it is doing what it does.

This means the strategy of action-expression described in Chapter 5 is more narratively friendly than action-selection. Instead of this emphasis on *selecting* the right action, Tom Porter suggests the strategy of *expressing* the reasons an agent does an action and the emotions and thoughts that underly its activity [Porter, 1997]. This means organizing the agent architecture so that reasons for behavioral change are explicit and continuously expressed. By showing not only what the agent does, but why the agent does it, people may have an easier time understanding what the agent is thinking and doing in general.

A deeper problem with current architectures is that ethologically-based models such as [Blumberg, 1996] presuppose that most of what an agent does is basically stimulus-response. As scientists, we are not interested in the vagaries of free will; we want to develop cause-effect rules to explain why animals do what they do when they do it. We intentionally adopt what Daniel Dennett might call a 'non-intentional stance' [Dennett, 1987]. We therefore develop theories of behavior that are fundamentally mechanistic.

But when we build agents that embody these theories, they often work through stimulus-response or straightforward cause-effect. This automaticity then carries forward into the quality of our agent's behavior.

As a consequence, agents are not only non-intentional for us; they are also reduced to physical objects in the eyes of the user. Narrative Intelligence requires agents that at least appear to be thinking about what they are doing and then making deliberate decisions, rather than simply reacting mindlessly to what goes on around them. We may be automatic; but we should not appear so.

4.Hermeneutic Composability

Narrative is understood as a type of communication between an author and an audience. In order to understand this communication, the audience needs to go through a process of interpretation. At the most basic level, the audience needs to be able to identify the 'atomic components' or events of the narrative. But this is just the beginning; the audience then interprets the events not in and of themselves but with respect to their overall context in the story. Once the story is understood, the events are re-identified and re-understood in terms of how they make sense in the story as a whole. In essence, this is a complex and circular process: the story only comes into being because of the events that happen, but the events are always related back to the story as a whole.

This property of narrative is another nail in the coffin of the dream of plug-n-play agents. If users continuously re-interpret the actions of the agent according to their understanding of everything the agent has done so far, then agent-builders who design the parts of their agents completely separately are going to end up misleading the user, who is trying to understand them dialectically.

More fundamentally, the deep and complex interrelationships between the things creatures do over time is part of what makes them come alive, so much so that when there are deep splits between the 'parts' of a person — for example, they act very happy when they talk about very sad things — we consider them mentally ill. This kind of deep consistency across parts is very difficult to engineer in artificial systems, since we do not have methodologies for engineering wholistically. It may be that the best we can do is the surface impression of wholism; whether that will be enough remains to be seen.

5.Canonicity and Breach

A story only has a point when things do not go 'the way they should.' "I went to the grocery store today" is not a story; but it is the beginning of a story when I go on to say "and you'll never believe who I ran into there." There is no point to telling a story where everything goes as expected; there should be some problem to be resolved, some unusual situation, some difficulty, someone behaving unexpectedly.... Of course, these deviations from the norm may themselves be highly scripted ("boy-meets-girl, boy-loses-girl, boy-wins-girl-back" being a canonical example).

It may be, then, that the impression of intentionality can be enhanced by making the agent do something unexpected. Terrel Miedaner's short story "The Soul of the Mark III Beast" revolves around just such an incident [Miedaner, 1981]. In this story, a researcher has built an artificially intelligent robot, but one of his friends refuses to believe that a robot could be sentient. This continues until he hands her a hammer and tells her to destroy the robot. Instead of simply breaking down — the friend's canonical expectation — the robot makes sounds and movements that appear to show pain and fear of death. This shakes the friend so much that she starts to wonder if the robot is alive, after all. Watching the robot visibly grapple with its end, the friend is led to sympathy, which in turn leads her to see the robot as sentient.

More generally, people come to agents with certain expectations, expectations which are again modified by what they see the agent do. The appearance of intentionality is greatly enhanced when those expectations are not enough to explain what the agent is doing. That is, the agent should not be entirely predictable, either at the level of its physical actions or at the level of its overall behavioral decisions. Characters in a Harlequin romance — who inevitably fall in love with the man they hate the most [James, 1998] — have nowhere near the level of 3-dimensionality of the complex and quirky characters of a Solzhenitsyn novel. Similarly, agents who always do the same thing in the same situation, whose actions and responses can be clearly mapped out ahead of time, will seem like the automatons they are, not like fascinating living creatures.

Since one of the goals of Narrative Intelligence is to make agents more naturally readable, stereotypicity may seem like a helpful step towards that goal. After all, if the agent always does the same thing for the same reasons in the same ways, the user will always know exactly what the agent is doing. But since users are very good at creating narrative, stereotyped actions bore the audience. In order to create compelling narrative, there needs to be some work for the reader to do as well. The agent designer needs to walk the line between providing enough cues to users that they can create a narrative, and making the narrative so easy to create that users are not even interested.

6.Referentiality

The 'truth' in stories bears little resemblance to scientific truth. The point of stories is not whether or not their facts correspond to reality, but whether or not the implicit reasoning and emotions of the characters 'feels' right. A plausible narrative does not essentially refer to actual facts in the real world, but creates its own kind of "narrative world" which must stand up to its own tests of 'reality.'

Similarly, extensive critiques have been made in AI about the problem of trying to create and maintain an objective world model [Agre, 1997]. Having the agent keep track of the absolute identity and state of objects in the external world is not only difficult, it is actually unhelpful. This is because in many situations the absolute identity of an object does not matter; all that matters is how the agent wants to or could use the object. As a substitute, Philip Agre has introduced the notion of 'deictic representation,' where agents keep track of what is going on, not in any kind of absolute sense, but purely with respect to the agent's current viewpoint and goals [Agre, 1988].

While understanding the power of subjectivity for agents, AI in general has been more reluctant to do away with the goal of objectivity for agent researchers. AI generally sees itself for better or for worse as a science, and therefore valorizes reproducibility, testability, and objective measures of success. For many, 'intelligence' is a natural phenomenon, independent of the observer, which is to be reproduced in an objective manner. Intelligence is not about appearance, but about what the agent 'actually' does. This reveals itself in the oft-repeated insistence that agents should not just appear but *be* 'really' alive or 'really' intelligent — anything less is considered illusionary.

This 'real' essence of the agent is usually identified with its internal code — which is also, conveniently enough, the AI researcher's view of the agent. As a consequence, as described in Chapter 5, the impression the agent makes on the user is often considered less real, and by extension, less important. This identification of the internal code of the agent as what the agent really *is* — with the impression on the user a pale reflection of this actual essence — has an unexpected consequence: it means that the subjective interpretation of the audience is devalued and ignored. The result is agents that are unengaging, incoherent, or simply incomprehensible.

This does not mean the AI community is idiotic. Most AI researchers simply have a scientific background, which means they do not have training in subjective research. But the accent on AI as a science, with the goals and standards of the natural sciences, may lose for us some of what makes narrative powerful. I do not believe that 'life' in the sense of intentionality will be something that can be rigorously, empirically tested in any but the most superficial sense. Rather, generating creatures that are truly alive will probably need to tap into the arts, humanities, and theology, which have spent centuries understanding what it means to be alive in a meaningful way. While intelligent tools may be built in a rigorous manner, insisting on this rigor when building our 'robot friends' may be shooting ourselves in the foot.

7.Genericness

Culturally-supplied genres provide the context within which audiences can interpret stories. Knowing that a story is intended to be a romance, a mystery, or a thriller gives the reader a set of expectations that strongly constrain the way in which the story will be understood. These genre expectations apply just as well to our interpretations of everyday experience. The Gulf War, for example, can be understood as a heroic and largely victimless crusade to restore Kuwait to its rightful government or as a pointless and bloody war undertaken to support American financial interests, depending on the typical genre leanings of one's political philosophy.[5].

These genres within which we make sense of the world around us are something we largely inherit from the culture or society we inhabit. This means at its most basic that different kinds of agent behavior make sense in different cultures. For example, I once saw a Fujitsu demo of 'mushroom people' who would, among other things, dance in time to the user's baton. In this demo, the user went on swinging the baton for hours, making the mushroom people angrier and angrier. Finally, it was the middle of the night, and the mushroom people were exhausted, obviously livid — and still dancing. I thought this behavior was completely implausible. "Why on earth are they still dancing? They should just leave!" I was told, "But in Japan, that would be rude!" My American behavioral genre

[5] A similar perspective is used to automatically generate ideologically-based understanding of news stories in [Carbonell, 1979] For a humanist example of the effect of generic ways of thinking on the actions we take in our everyday lives, see [Sontag, 1979].

expectations told me that this behavior was unnatural and wrong — but in Japan the same behavior is correct.

Since cultural expectations form the background within which agent behavior is understood, the design of intentionally comprehensible agents needs to take these cultural expectations into account. Patricia O'Neill-Brown points out that this means the current practice of building agents without thinking about the specific cultural context in which the agent will be used is likely to lead to agents that are misleading or even useless [O'Neill-Brown, 1997]. This means an understanding of the sociocultural environment in which an agent will be inserted is one important part of the agent design process. In fact, O'Neill Brown goes one step further: not only does cultural baggage affect the the way agents *should* be designed, it *already* affects the way agents are designed. That is to say, the way designers think of agents has a strong influence on the way we build them to start out with.

This should not come as a surprise to readers of this thesis. In Chapter 1, we already saw how classical and alternative AI work on metaphors of agenthood that are more broadly operative in culture. AI research itself is based on ideas of agenthood we knowingly or unknowingly import from our culture. Given that this is the case, our best bet for harnessing the power of culture so it works for AI instead of against it is the development of 'critical technical practices,' including a level of self-reflective understanding by AI researchers of the relationship between the research they do and culture and society as a whole [Agre, 1997].

8.Normativeness

Previously, we saw that a story only has a point when things do not go as expected; similarly, agents should be designed so that their actions are not completely predictable. But there is a flip side to this insight: since the 'point' of a story is based on a breach of conventional expectations, narratives are strongly based on the conventions that the audience brings to the story. That is, while breaking conventions, they still depend on those same conventions to be understood and valued by the audience.

Intentional agents, then, cannot be *entirely* unpredictable. They play on a tension between what we expect and what we do not. There needs to be enough familiar structure to the agent that we see it as someone like us; it is only against this background of fulfilled expectations that breached expectation comes to make sense.

9.Context Sensitivity and Negotiability

Rather than being presented to the reader as a *fait accompli*, narrative is constructed in a complex interchange between the reader and the text. Narrative is assimilated by the reader based on that person's experiences, cultural background, genre expectations, assumptions about the author's intentions, and so on. The same events may be interpreted quite differently by different people, or by the same person in different situations.

In building narrative agents, on the other hand, the most straightforward strategy is context-free: (1) decide on the default narrative you want to get across; (2) do your best to make sure the audience has understood exactly what you wanted to say. The flaw in this strategy is that narrative is not 'one size fits all.' It is not simply presented and then absorbed; rather, it is constructed by the user. In assimilating narrative, users relate the narrative to their own lived experience, organizing and understanding it with respect to things that have happened to them, their generic and conventional expectations, and their patterns of being. Narrative is the interface between communication and life; through narrative a story becomes a part of someone's existence.

This means the 'preformed narrative' that comes in a box regardless of the audience's interests or wishes is throwing away one of the greatest strengths of narrative: the ability to make a set of facts or events come to life in a meaningful way *for the user* — in a way that may be totally different from what someone else would see. Rather than providing narrative in a prepackaged way, it may be more advantageous to provide the *cues* for narrative, the building blocks out of which each user can build his or her unique understanding.

And if narrative is not the same for everyone, then narrative agents shouldn't be, either. If narrative is fundamentally user-dependent, then inducing narrative effectively means having some ideas about the expected audience's store of experience and typical ways of understanding. Just as the author of a novel may have a typical reader in mind, the designer of an agent needs to remember and write for the users who will be using that agent, relating the agent's projected experiences to the lived experience of the desired audience.

And just as the author of a novel does not expect every possible reader to 'get the point,' the author of an agent does not necessarily need to be disappointed if only some people understand what the agent is about. The statistical testing of an agent's adequacy over user population may miss the point as much as using bestseller lists to determine the quality of novels. It may be that making the point well with a few users is better, from the point of view of the designer, than making the point adequately with many users.

10.Narrative Accrual

Generally speaking, narratives do not exist as point events. Rather, a set of narratives are linked over time, forming a culture or tradition. Legal cases are accumulated, becoming the precedents that underly future rulings. Stories we tell about ourselves are linked together in a more-or-less coherent autobiography.

The mechanism by which narratives accrue is different from that of scientific fact. We do not find principles to derive the stories, or search for empirical facts in the stories to accept or reject according to a larger paradigm. Stories that contradict one another can coexist. The Bible, for example, first cheerfully recounts that, on the 7th day, God made man and woman at the same time; a little later, God makes man out of mud, and only makes woman after man is lonely [Various, 1985]. We don't necessarily have a problem reconciling two stories, in one of which Fred is mean, and in the other he is nice. The process of reconciliation, by which narratives are joined to create something of larger meaning, is complex and subtle.

The ways in which stories are combined — forming, if not a larger story, at least a joint tradition — is not currently well-understood. Once we have a better understanding of how this works, we could use these mechanisms in order to modulate the effects of our narrative agents as they move from episode to episode with the user. As Dautenhahn has suggested, agents are understood by constructing 'biographies' over the course of prolonged interaction. By investigating the mechanisms whereby the user constructs these biographies from the mini-narratives of each encounter, we stand a better chance of building our agent so that it makes the desired effect on the user.

Narrative and Atomization

In general, narrative involves understanding the wholistic relationships between things: the relationship between the different events in the story, the relationship between the events and how the actors feel about the events, the relationship between what the author tries to tell and the way in which the audience constructs what it hears, the relationship between the audience member and his or her cultural background, and so on. With layer upon layer of interdependency, this narrative view of the world can become extremely complex.

In contrast, the scientific worldview tends to value simplicity through black-boxing, our old friend atomization. As a reminder, atomization is the process of splitting something that is continuous and not strictly definable into reasonably well-defined, somewhat independent parts. We do this for a good reason: atomization is a way of getting a handle on a complex phenomenon, a way of taking something incoherent, undefined, and messy and getting some kind of fix on it. It is only through atomization that we can understand something clearly enough to be able to engineer a working system of any complexity. Atomization is essential to AI.

But atomization as used in science is not a transparent methodology. In many ways, its properties are the exact opposite of those of narrative. This can be seen more concretely by inverting each of the properties of narrative:

1. *Structure over time*: Narrative structure is diachronic; it is about how events relate to one another. Atomistic structure is statistical. Patterns of events over time are simply correlated with one another.

2. *Essence*: Narrative is interested in particular events; it matters which person a story is about. Atomization is interested in finding salient properties so that events can be generalized as parts of a system; individual water molecules, for example, are not differentiated. Narrative sees events as essentially particular; atomization, as essentially abstract, with specific features seen as 'noise.'

3. *Components*: Narrative is interested in events mainly in terms of how the actors involve understand and interpret them. Scientific atomization is interested in the facts that can be established independently of any one person's experience.

4. *Combination*: Narrative is wholistic; the act of bringing its components together changes the components themselves. In atomization, the combination of events is seen as the sum of the parts.

Aspect	Scientific worldview	Humanist worldview
structure over time	statistical	diachronic
essence	abstract	particular
components	factual	experiential
combination	additive	wholistic
relation to expectation	predictable	creative
referentiality	objective	subjective
dependence on culture	culturally universal	culturally variable
audience judgment	unimportant	essential
application	absolute	context-sensitive
accrual	logical coherence	tradition

FIGURE 6.1: Relations between scientific (atomistic) and humanist (narrative) worldviews

5. *Relation to expectation*: Narrative must contain elements that are unexpected; things cannot go as planned. In contrast, the goal of scientific atomization is to be able to predict and control with reasonable certainty the outcome of events.

6. *Referentiality*: Narrative is fundamentally subjective; it is about how different people come to interpret it in different situations. Scientific atomization is meant to be objective. Its laws hold in every situation, independent of context and interpretation.

7. *Dependence on culture*: Similarly, while narrative is largely dependent on culturally bound norms and expectations, scientific atomization is thought of as culturally universal, true for everyone.

8. *Audience judgment*: The audience must use its judgment for narrative to be realized; but audience judgment is considered to be unimportant for determining the truth of scientific atoms.

9. *Application*: The way in which narrative is used depends on context; atomic facts are meant to be absolute.

10. *Accrual*: Narratives are combined to form a not necessarily coherent tradition. Atomic facts are combined by comparing them and finding a logical structure that subsumes them. Facts that are inconsistent are thrown away.

These aspects are summarized in Figure 6.1.

Clearly, these statements are too absolute. Not all scientific work is, for example, interested purely in statistical properties of events. Many forms of science have shaded over to the narrative end of the spectrum. Psychiatry and neurology, for example, often depends heavily on case studies, which chronicle the particular life history of individual patients. While science, being a heterogeneous set of practices, cannot be absolutely identified with the purely atomistic end of the spectrum, scientific values and practices do cluster towards atomization. Similarly, the humanities are not unanimous in being placed at the purely narrative end, but humanistic projects do tend to have more of the narrative attributes.

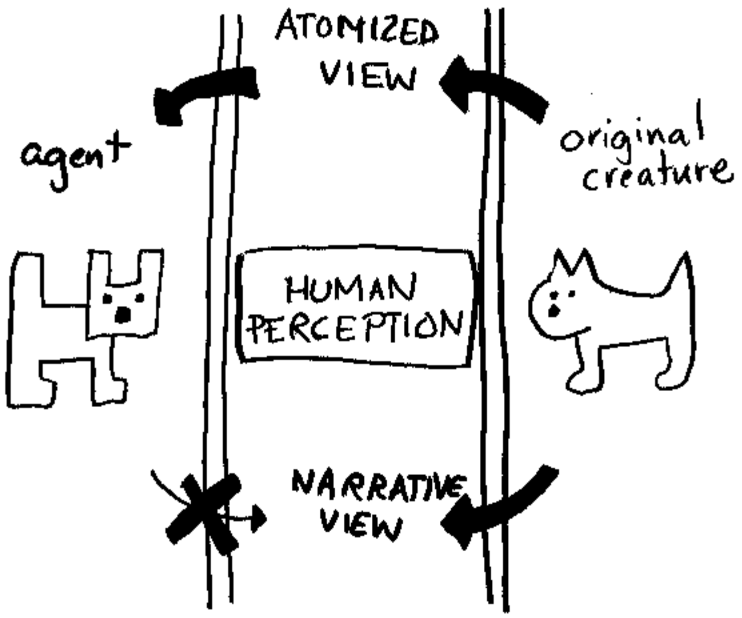


FIGURE 6.2: An atomized creature is likely to be narratively incomprehensible

This means the division of atomization from narrative is meaningful, at least heuristically.

Atomization, Narrative, and AI

Atomization is an essential tool for AI:

> There are many possible approaches to building an autonomous intelligent system. As with most engineering problems they all start by decomposing the problem into pieces, solving the subproblems for each piece, and then composing the solutions.[Brooks, 1986b]

But because atomization is closely linked with mechanicity, its value must be called into question when the goal is building truly intentional beings. As narrative psychology has demonstrated, when humans try to make intentional behavior meaningful, they use a fundamentally different procedure from that of atomization and the scientific method. Rather, humans create meaning by structuring their experience according to narrative, in the tradition of the humanities. This difference between the atomistic standpoint of the agent designer and the narrative viewpoint of the eventual agent audience can undermine the designer's ability to construct intentionally understandable agents.

To understand how this works, consider Figure 6.2. On the right is the living agent - or idea of an agent — that the designer wants to copy. The designer tries to understand the dynamics of this agent's behavior by finding out its atomic constituents. For example, the designer may try to find out the typical activities in which the agent engages, the conditions under which each activity is likely to occur, and the length of time the agent tends to spend on various activities. Using these facts, the designer can construct a system that has the same attributes. Once the system can generate behavior that closely approximates the finite list of atomic attributes with which the designer has measured the agent, the designer is satisfied that the agent is a reasonable facsimile of the living agent. Scientifically speaking, the designer is correct.

But now consider the user's point of view. Rather than being interested in the empirically determinable individual attributes of the creature, the user focuses on how the creature's activities seem to meld together into a whole. The narrative attributes of the agent's activities — the extent to which the agent's behavior is not simply the sum of predictable parts — is precisely what the scientific copy of the creature has left out. This means that *even if the designer succeeds in making an accurate copy according to scientifically measurable properties, from the point of view of the user the living creature is fundamentally different from the constructed agent*.

If we are to build agents that truly appear intentional, then, we need to include narrative properties in our design of artificial creatures. Currently, many (though by no means all) AI techniques fall on the 'scientific' end of the spectrum in Figure 6.1. This atomistic worldview reflects itself not only in the internal code of the agents, but also in the quality of the externally observable behavior that forms the basis by which audiences try to understand the agent. The challenge for an AI that wants to build, not just intelligent tools, but intentional agents, is to find ways of moving AI methodology towards the values embodied in narrative. The point is not that narrative is good and science as embodied in current AI is bad, but that we need *both* narrative and AI techniques set in relationship to one another. In the next chapter, we will explore these possibilities through the structure of the Expressivator, an AI architecture that embodies many narrative principles.

Chapter 7

Architectural Mechanisms II: Transitions as Narrative

In Chapter 2, we defined schizophrenia as a deficiency in agent behavior integration characterized by short dalliances in individual behaviors with sharp breaks between behaviors. This schizophrenia has its origins in the reduction of the overall dynamics of agent activity to crystallized atomic behaviors. This led to the hypothesis in Chapter 5 that we could address schizophrenia with transitions. These transitions would cover over the breaks between behaviors, so they would be less noticeable to users.

But animation and narrative psychology suggest that the fundamental problem with current agent-building techniques is not simply recognizable atomization in and of itself, but rather that atomized agents do not provide proper support for narrative interpretation. Abrupt behavioral breaks create the (often correct) impression that there is no relationship between the agent's behaviors; rather than focusing on understanding the agent as a whole, the user is left to wonder how individually recognizable behaviors are related to each other and the agent's personality. Behaviors are designed in isolation and interleaved according to opportunity — but users, like it or not, attempt to interpret behaviors in sequence and in relationship to each other. The result of this mismatch between agent design and agent interpretation is confusion on the part of the user and the likelihood that the designer's conception of the agent will be miscommunicated.

If we want to solve these problems of miscommunication, it may be better to use transitions, not simply to cover up splits in the agent's construction, but to provide cues for users to construct narrative. This means that transitions should not smooth together but *relate* atomic behaviors, explaining to users the reasons behind the agent's behavioral changes. Instead of simply hiding the problems of atomization by blending behaviors together, transitions as narrative express to the user what the agent is thinking and doing.

In Chapter 5, I described mechanisms for the Expressivator that were based on the idea of agent as communication and transitions as behavior-blending. My initial goal was to simply add transitions to Hap, the behavior-based architecture from which the Expressivator was developed, as a 'glue' between Hap's behaviors. It turned out, however, that using transitions well necessitated some basic changes in the way the Expressivator is used.

Here, I will describe the Expressivator as it emerged from this research. The notion of agent as communication is still crucial, but over time it became clear that it was more useful to think of transitions as support for *narrative* rather than as behavior-blending. This re-thought Expressivator, as I will describe in gorey detail in this chapter, is therefore based on the concept of agent-as-narrative. This use of transitions led to a substantial re-understanding of the nature of Hap's default architectural mechanisms. In this chapter, I will explain the structure of the final Expressivator, how it was used to implement the Patient in the Industrial Graveyard, how it changes the nature of Hap as an agent programming language, its limitations, and what it could lead to in the future.

In this chapter there is a distinct tension between the need to give enough technical details to make technical readers feel they fully understand the system and the hope that humanist readers will not be entirely lost under a barrage of technical verbiage. I have therefore kept the body of this chapter relatively straightforward, moving more technical sections to the appendix. Technical readers may want to interlace their reading of this chapter with the appropriate sections of the appendix; the sections to read at each point are pointed out in the text.

Expressivator as Support for Narrative Comprehension

The fundamental change that was required in order to make the Expressivator function effectively to support narrative comprehension is this:

This is the fundamental technical point of this thesis.

> Behaviors should be **as simple as possible**. The agent's life comes from thinking out the **connections** between behaviors and **displaying** them to the user.

This is the concrete, technical manifestation of what it means to be a narratively expressive agent.

This heuristic is in some sense simply restating the point of making agents expressive. But it turns out to have extensive ramifications on technical practice. Most specifically, it forced me to go against my natural tendency in behavior-building: to try to create the appearance of lifelike complexity in the behavior of the agent by making the actual code of the agent extremely intricate. This internal complexification certainly does make the agent's actions more complex, but it does not make the agent seem more intentional. In my experience, the *only* thing that really makes the agent seem intentional is the addition of clear reactions and behavioral sequences that show the agent thinking about what is going on around it.

Simpler behaviors are essential because *complex processing is lost on the user*. Most of the time, the user has a hard time picking up on the subtle differences in behavior which bring such pleasure to the heart of the computer programmer. But the properties of narrative interpretation mean that simpler behaviors are also *enough*. Because the user is very good at interpretation, *minimal behavioral cues suffice*. The *signifiers* of Chapter 5 become these simple behaviors here, focusing on the cues (or, technically speaking, *signs*) which communicate the desired behavior to the user.

For narrative understanding, users are not simply interested in what the agent is doing from moment-to-moment, but in how the agent's actions relate to each other over time. Specifically, they do not just want to know *what* the agent is doing, but *why*. The Expressivator uses transitions, not to smooth between behaviors as in Chapter 5, but to *express the reasons for the agent's behavioral choices*. Transitions do not hide behavioral change, but instead make clear the *reasons* for it and the *relationships* between the agents' behaviors. These transitions are, as in Chapter 5, implemented using meta-level controls.

The reader has already been introduced to the mechanisms of signifiers, transitions, and meta-level controls. In this chapter, I will discuss the use of these mechanisms within the context of Narrative Intelligence. Signifiers and meta-level controls remain more or less the same, but *transitions* are altered, both in implementation and in use. Transitions now focus on the *reasons* for behavioral change; they are implemented using *transition triggers*, which note when change for a particular reason is necessary, and *transition demons*, which express that reason to the user. After we briefly revisit signifiers and look at transitions in more detail, I will return to look at how the entire process of agent design changes under the Expressivator, because of its focus on the presentation of behavioral interrelationships.

Signs and Signifiers Reviewed

As described in Chapter 5 (pp. 113-121), behaviors are hierarchized according to their level of meaning-generation. At the lowest level, behaviors are built out of physical and mental actions. Physical and mental actions are combined to create context-sensitive *signs*, which are the lowest level at which the agent's behavior communicates to the audience.

Actions and signs are in turn combined to generate low-level signifiers. Low-level signifiers are relatively simple behaviors that convey a particular kind of activity to the user. The Patient's behaviors include such low-level signifiers as "react to the Overseer," "look around curiously," or "sigh." Unlike low-level behaviors in other systems, which may or may not be noticed by the user, low-level signifiers are explicitly intended to be communicated; users should be able to identify the low-level signifiers more or less correctly.

Low-level signifiers are combined to build up high-level signifiers. High-level signifiers are collections of low-level signifiers that together form a complex, high-level activity, such as "explore the world" or "mope by the fence." The high-level signifiers in turn combine to create the full behavior of the agent. The high-level signifiers used to create the Patient are shown along with the low-level signifiers they contain in Figure 7.1.

High-level Signifier	Low-level Signifiers
In Monitor	Act mechanical Tremble and watch overseer Look around scared Look around curiously
Explore World	Go to spot Examine spot Look around Sigh React to Overseer
Read Sign	Read line React to line
Exercise	Bob up and down
Turned off	Stay turned off
Mope by Fence	Look out at world Sigh Walk up and down fence
Head-Banging	Hit head on ground Wait to see if light goes out Act frustrated
Be Killed	Act afraid Die

FIGURE 7.1: High-level ahd Low-level Signifiers in the Patient

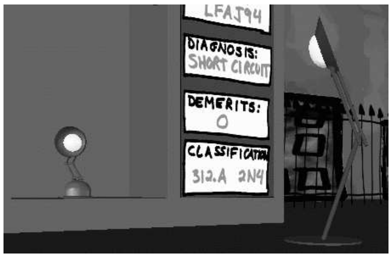


FIGURE 7.2: The Patient, scanning the junkyard mechanically.

Transitions

Transitions are used in order to relate atomic behaviors to one another. Transitions explain to the user why the agent is moving from one kind of behavior to another. Since there are two kinds of behaviors, there are also two kinds of transitions, though they are implemented in analogous ways: mini-transitions and maxi-transitions.

'Mini-transitions' connect low-level signifiers to form high-level signifiers. For example, when being examined in the monitor, the Patient initially acts lifelessly. It scans the environment slowly, doing its best to look mechanical (Figure 7.2). When the Patient notices the Overseer, the Patient suddenly comes to life, trembling and following the movements of the Overseer nervously (Figure 7.3). This change in the Patient is reinforced through a mini-transition that displays a shock reaction and

FIGURE 7.3: The Patient trembling and watching the Overseer.

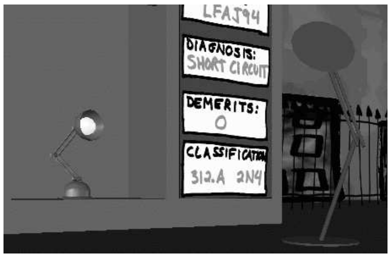


FIGURE 7.4: Shock reaction

backs up from the Overseer (Figures 7.4- 7.5). These simple movements draw more attention to the Patient's reaction to the Overseer, thereby encouraging the user to understand that a palpable change has happened to the Patient, triggered by the presence of the Overseer.

'Maxi-transitions' connect high-level signifiers in order to create the agent's overall activity. When the Patient changes from moping at the fence (Figure 7.6) to headbanging, the maxi-transition first turns its head to the camera (Figure 7.7) so the user can see the Patient's light going out (Figure 7.8). Then, the Patient shakes its head a few times (Figures 7.9- 7.10), with the light flashing on and off (Figure 7.11). Hopefully, by the time the Patient begins to hit its head on the ground (Figures 7.12- 7.13), the user has understood that something is wrong with the Patient's light and that the headbanging behavior is intended to fix the short circuit, not to hurt itself.[1]

[1] In practice, this behavior is still not entirely clear, for reasons to be explained later.

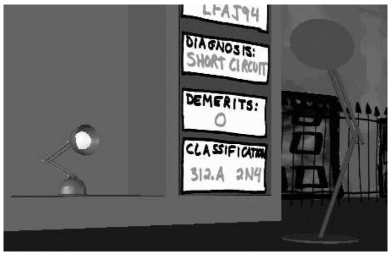


FIGURE 7.5: The Patient scoots back from the Overseer

FIGURE 7.6: The Patient moping by the fence

FIGURE 7.7: The Patient, sadly bringing its lightbulb into full view.

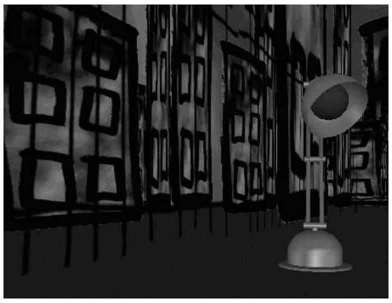


FIGURE 7.8: The user can see the light going out

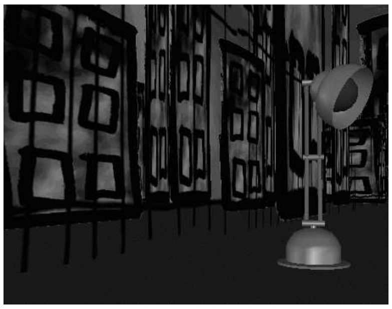


FIGURE 7.9: Shaking head, movement 1

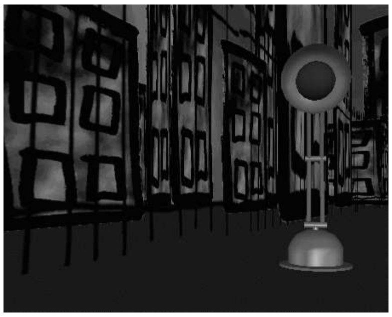


FIGURE 7.10: Shaking head, movement 2

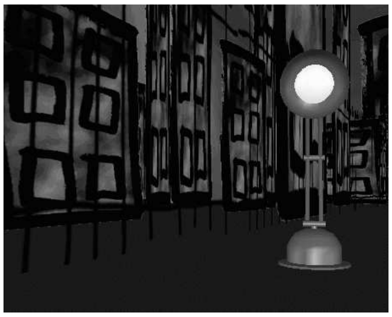


FIGURE 7.11: The lightbulb flashes

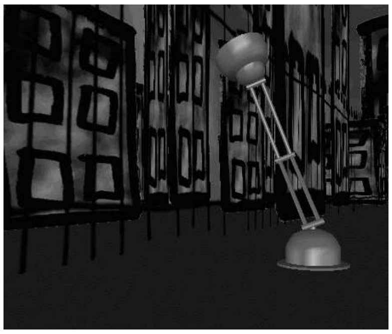


FIGURE 7.12: Headbanging starts

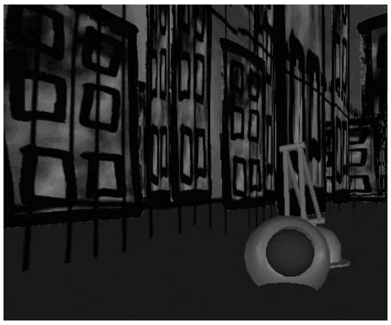


FIGURE 7.13: Headbanging in full gusto

Transition Implementation

Conceptually, transitions are intended to communicate the reason an agent is switching from one behavior to another. But for each reason an agent has for switching, there may be more than one way of communicating that reason, depending on local contextual conditions. For example, whenever the Patient notices the Overseer coming nearby, it switches from whatever it is doing into a defensive mode. The reason for this change is that the Patient is frightened out of its wits by the Overseer. Usually, the correct way to communicate this fear is to have the Patient whirl around, face the Overseer, and start cowering. But when the Patient's light is out, it cannot see, so it would be inappropriate to communicate fear by having the Patient look at the Overseer. Instead, when it 'hears' the Overseer approach, it whirls around frantically, trying to figure out where the Overseer is. So, depending on whether or not the Patient can see, there are two ways of actually showing the user that the Patient is switching behaviors out of fear of the Overseer.

In order to allow for this disjunction between the reason for a behavioral change and the appropriate communication of that reason, transitions are implemented in two parts: (1) *transition triggers*, that determine when it is appropriate to switch to another behavior, and (2) *transition demons*, that implement the transition sequence itself. The transition trigger notes when a particular reason for behavioral change has been fulfilled. It generally uses the *sensing behaviors* meta-level control in order to find out which behaviors are running (e.g. exploring the world), and combines this information with sensory input (e.g. the Overseer is approaching). The transition demon figures out how to communicate that reason for change to the user, according to the current history of user-agent interaction and other conditions in the virtual environment. The reason is expressed behaviorally with the help of the full range of meta-level controls described in Chapter 5.

The technical reader is now referred to section D.1 of the Appendix for more fascinating information on transition implementation.

Transitions and What They Communicate: Two Case Studies

The best way to understand how transitions change the quality of agent behavior is to look at some of them in detail. Here, I'll go over two points where the agent switches behaviors, and explain what it looks like both without and with transitions. These case studies should help give a feel for the kinds of things transitions can help communicate to the user.

Reading the Schedule to Exercising

Towards the middle of the story, the Patient notices the schedule of daily activities which is posted on the fence. It goes over to read the schedule. The Overseer, noticing that the Patient is at the schedule and that the user is watching the Patient, goes over to the schedule, changes the time to 10:00, and forces the Patient to engage in the activity for that hour: exercising.

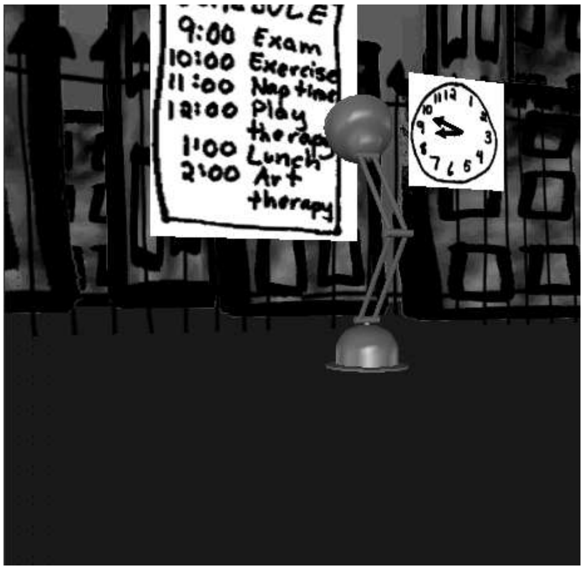


FIGURE 7.14: The Patient blithely reads the schedule...

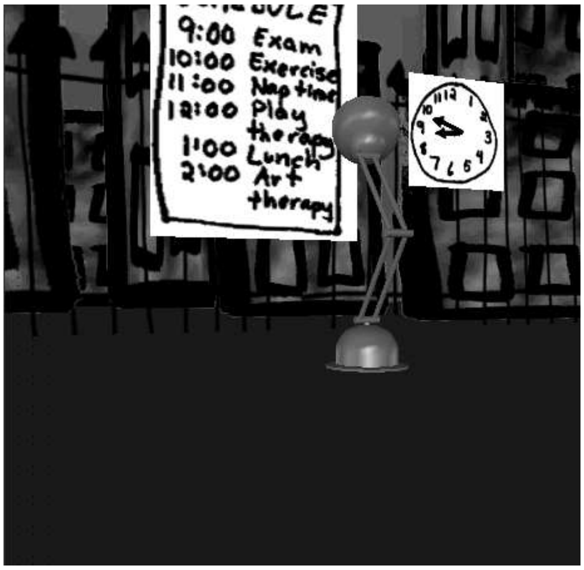


FIGURE 7.15: ... unmindful of the doom that awaits.

The goal of this part of the plot is to communicate to the user the daily regime into which the Patient is strapped. Being institutionalized, the Patient does not have autonomy over its actions; it can be forced by the Overseer to engage in activities completely independently of its desires. The specific behavioral change from reading the schedule to exercising, then, should show the user that the agent changes its activity because (1) it notices the Overseer, (2) the Overseer enforces the scheduled activities; (3) the activity that is currently scheduled is exercising.

Without transitions, the Patient's response to the Overseer is basically stimulus-response. The Patient starts out reading the schedule

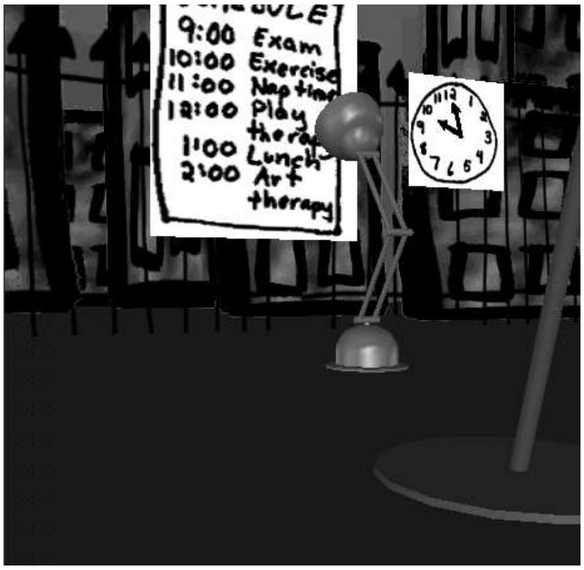


FIGURE 7.16: The Overseer approaches.

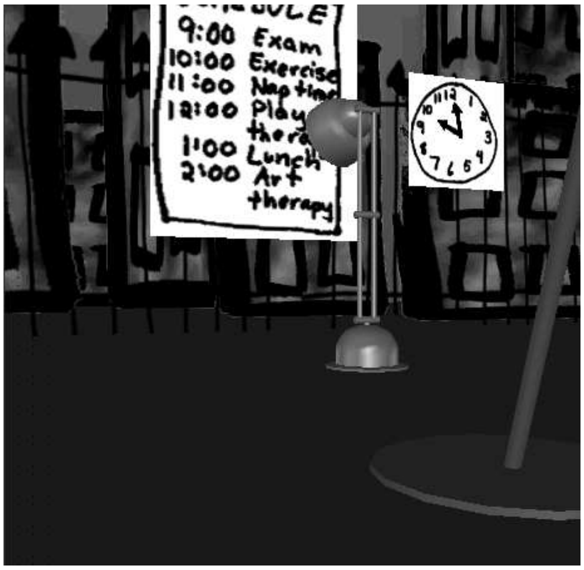


FIGURE 7.17: The Patient immediately begins exercising.

FIGURE 7.18: Exercising continues.

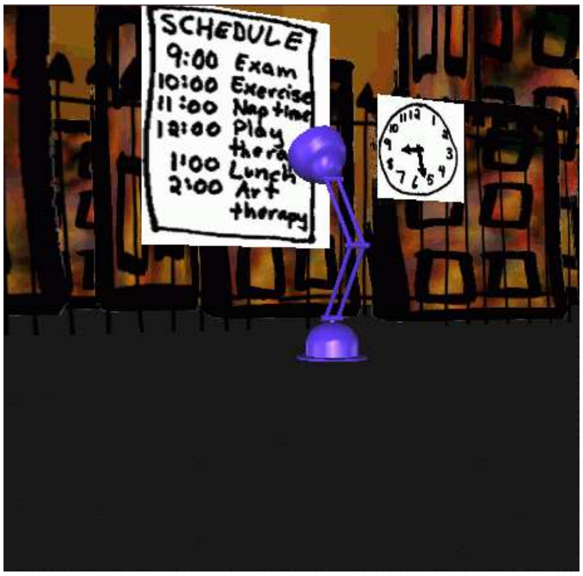


FIGURE 7.19: The Patient reads.

(Figures 7.14-7.15). As soon as the Patient senses the Overseer (Figure 7.16), it immediately starts exercising (Figures 7.17-7.18). This reaction is both correct and instantaneous; the Patient is doing an excellent job of problem-solving and rapidly selecting optimal behavior. But this behavioral sequence is also somewhat perplexing; the chain of logic that connects the Overseer's presence and the various environmental props to the Patient's actions is not displayed to the user, being jumped over in the instantaneous change from one behavior to another.

With transitions, attempts are made to make the logic behind the behavioral change more clear. Again, the behavior starts with the Pa-

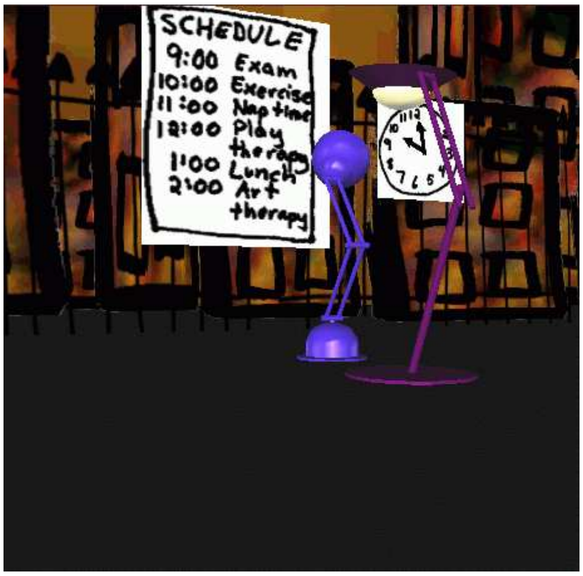


FIGURE 7.20: The Overseer approaches

FIGURE 7.21: The Patient lazily glances at the Overseer...

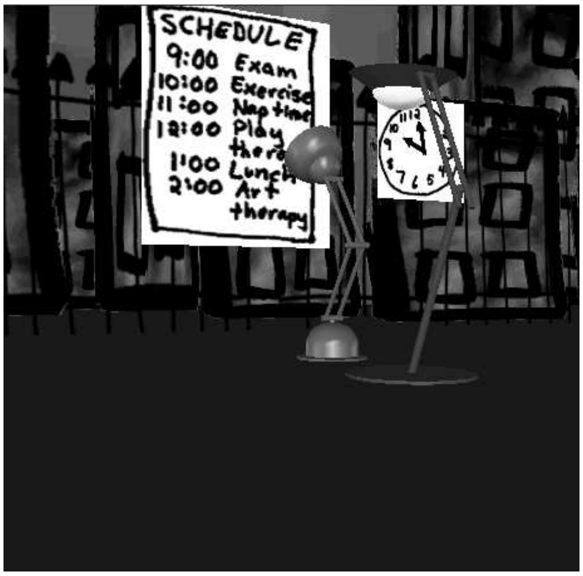


FIGURE 7.22: And returns to the far more interesting task of reading

FIGURE 7.23: Suddenly, the Patient has a heart attack

tient reading the schedule (Figure 7.19). This time, when the Overseer approaches (Figure 7.20), the Patient just glances at the Overseer (Figure 7.21) and returns to reading (Figure 7.22). Since the Patient normally has a strong fearfully reaction to the Overseer (and by this time the Overseer's enthusiasm for turning the Patient off has already generally aroused sympathy in the user's mind), the user has a good chance of understanding that this simple glance without further reaction means that the Patient has not really processed that the Overseer is standing behind it.

Suddenly, the Patient becomes startled (Figure 7.23) and quickly looks back at the Overseer again (Figure 7.24). Now, the user can get the

FIGURE 7.24: And looks back to confirm that the Overseer is there

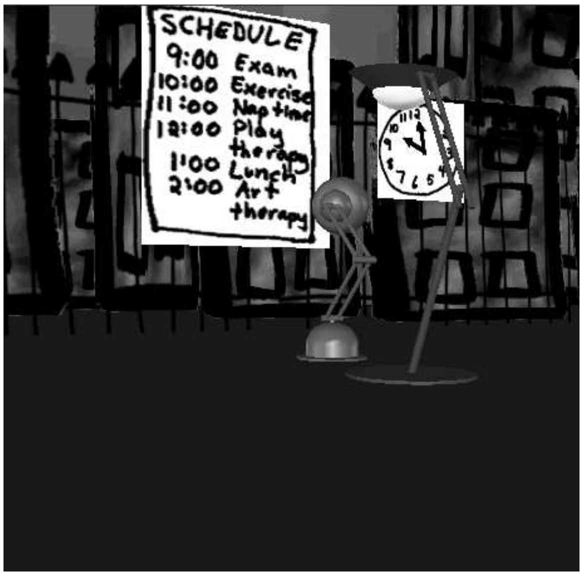


FIGURE 7.25: The Patient checks the time

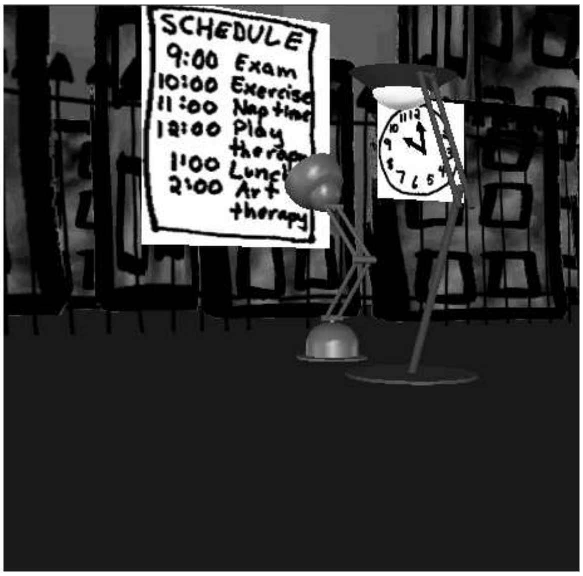


FIGURE 7.26: And checks the schedule to see what it should be doing

FIGURE 7.27: The Patient whirls to face the Overseer

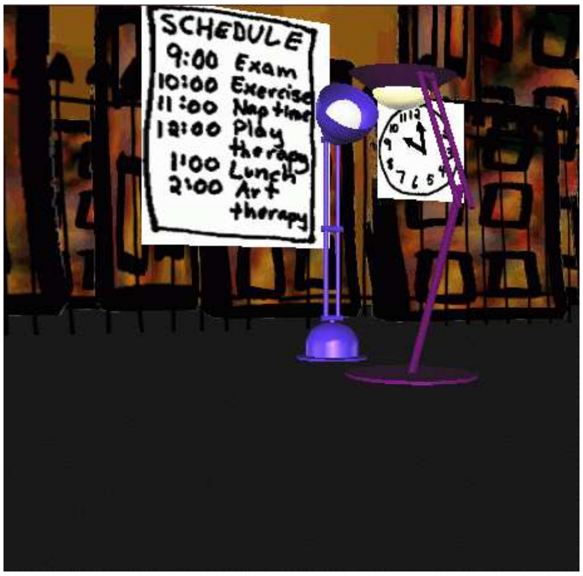


FIGURE 7.28: ... and frantically begins exercising

FIGURE 7.29: while staring at the Overseer

impression that the Patient has registered the Overseer's presence. Whatever happens next must be a reaction to that presence. Next, the Patient checks the time (Figure 7.25) and the schedule of activities (Figure 7.26) to determine that it is time to exercise. Then the Patient whirls to face the Overseer (Figure 7.27) and frantically and energetically begins exercising (Figures 7.28-7.29), tapering off in enthusiasm as the Overseer departs.

In practice, the timing on the animation is not quite right, so that users do not always interpret each substep of the transition correctly (this problem will be addressed below). Nevertheless, this transition clearly communicates that the change in behavior is connected to several factors: the presence of the Overseer, the clock, and the schedule. This is in contrast with the transition-less sequence, in which there is no clear connection between any of the environmental factors and the Patient's behavioral change.

Headbanging to Dying

Towards the end of the simulation, the Patient is frantically hitting its head against the ground, trying to fix its short circuit in the time-honored manner of the engineer. Because the headbanging movement involves the rapid motion of most parts of the Patient, it is also maximally bad; but the Patient itself is too worried about its lack of sight to worry about how good it is being. At this point the Overseer, who after numerous punishments has had its fill of monitoring the Patient, decides it is no longer efficient to allow the Patient to remain active. The Overseer comes over, maneuvers the Death Ray Machine over the Patient, which sends down a beam, turning the Patient into a lifeless 2-D texture map like the other junk in the junkyard.

At this stage of the game, I would like to communicate to the user that this is not just another temporary turn-off situation. What the Overseer is about to do is far worse than what it has done so far. In addition, this is my last chance to make the user feel guilty for his or her complicity in the scenario. The behavioral change from headbanging to death should make clear the horror of the situation, and be maximally guilt-inducing.

Without transitions, the scene proceeds in the following manner. As we join our character, we find it frantically whacking its head against the ground (Figures 7.30-7.31). As the Overseer approaches, the Patient instantly changes to the deathly fear behavior, which consists mostly of cowering and trembling (Figure 7.32). The Patient continues in this same behavior as the Overseer prepares for, and causes, its death (Figures 7.33-7.36).

Again, this behavioral change, while correct and somewhat effective, does not communicate to the User the full scale of what is going on. There is nothing in the Patient's behavior — who after all has been cowering and fearful for most of the story — that really points out that in this situation, something *really* bad is happening. The user probably does not have any inkling about the implications of the lowering of the Death Ray Machine until after it has done its dirty deed (one user, for example, thought it was merely an x-ray machine). Finally, while the user may feel sad for the Patient, there is nothing to make the user aware of the role he or she unwittingly has played in causing this behavioral change: the Patient's death.

FIGURE 7.30: The Patient is frantically headbanging

FIGURE 7.31: Whack, whack

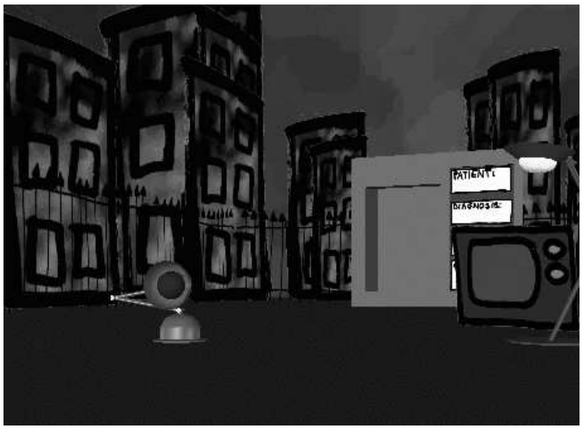


FIGURE 7.32: The Overseer approaches; instantly, the Patient freezes and trembles

FIGURE 7.33: The Patient continues to tremble as the Overseer lowers the death ray machine

FIGURE 7.34: ... and lowers it some more...

FIGURE 7.35: ... and as the Patient is zapped by the Death Ray

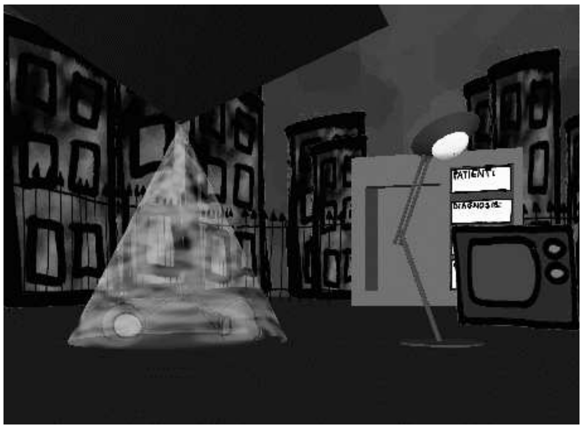


FIGURE 7.36: ... until the Patient finally dies.

FIGURE 7.37: Once again, we join the Patient as it is hitting its head

With transitions, these aspects of the Patient's behavioral change from headbanging to dying is made more clear.[2] Again, we start with the Patient hitting its head (Figures 7.37- 7.38). This time, when the Overseer approaches (Figure 7.39), the Patient crouches (Figure 7.40), and begins whirling around, trying to see where the Overseer is (Figures 7.41-7.43).

When the Death Ray Machine approaches, the Patient turns to face the camera, and therefore by extension the user (Figure 7.44); as the user watches, the Patient's light comes on (Figure 7.45). The Patient then slowly moves its gaze upwards toward the machine (Figure 7.46); when it sees the machine it starts trembling and quickly turns to the user (Figure 7.47). In case the user missed the implications of this move, the Patient repeats the sequence (Figures 7.48-7.49). The Patient's gaze then remains fixed on the user (Figure 7.50) as it continues to tremble until the sad end (Figure 7.51).

Experience with showing this sequence to users suggests that while the transition-less change is understandable, the sequence with transitions elicits both a better understanding of what is going on and a far greater sense of pity. The slow, trembling glances at the machine attract the user's attention to it; the user usually gets a good idea that something very bad

[2] I am extraordinarily grateful to Michael Mateas for helping me design this transition.

FIGURE 7.38: ... not realizing the sinister implications of what is about to happen

FIGURE 7.39: The Overseer approaches

FIGURE 7.40: The Patient crouches...

FIGURE 7.41: and whirls around blindly,...

FIGURE 7.42: trying to figure out where the Overseer is

FIGURE 7.43: (More whirling and trembling)

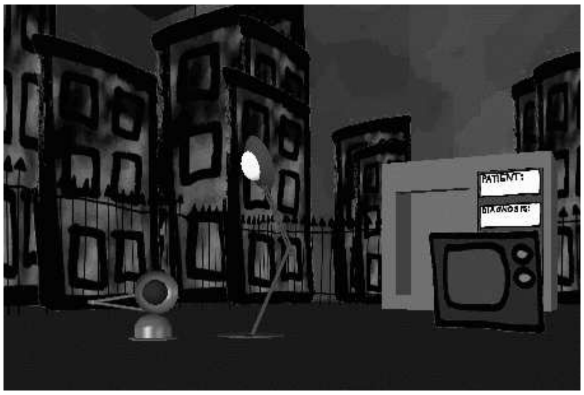


FIGURE 7.44: As the Death Ray machine comes overhead, the Patient turns to the camera.

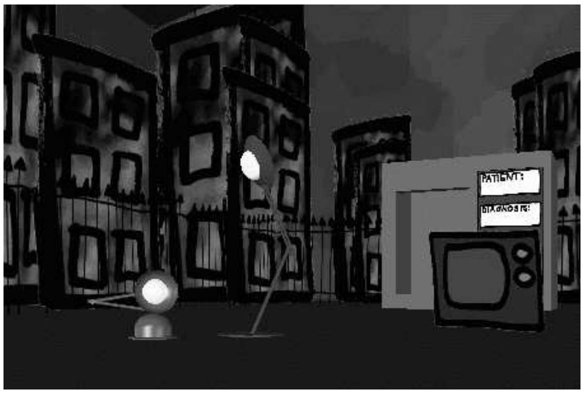


FIGURE 7.45: Its light comes on.

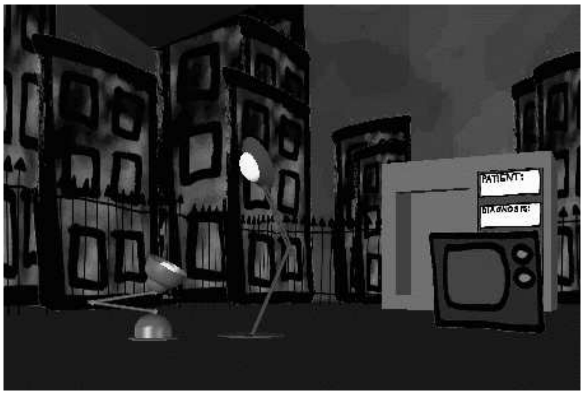


FIGURE 7.46: Slowly, the Patient turns its gaze up to the machine

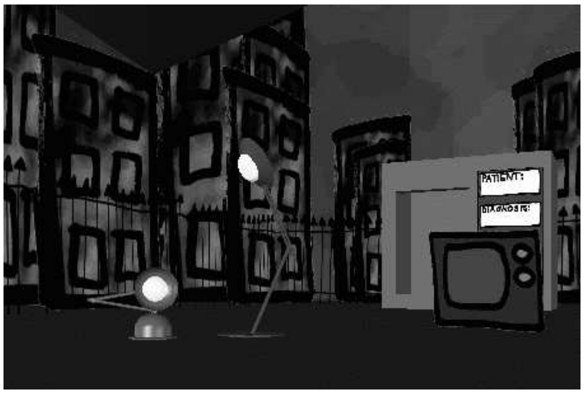


FIGURE 7.47: And looks at the camera, visibly trembling

FIGURE 7.48: Again, the Patient slowly looks up

FIGURE 7.49: And returns its gaze to the user while it trembles

FIGURE 7.50: Its gaze remains on the user as it is zapped

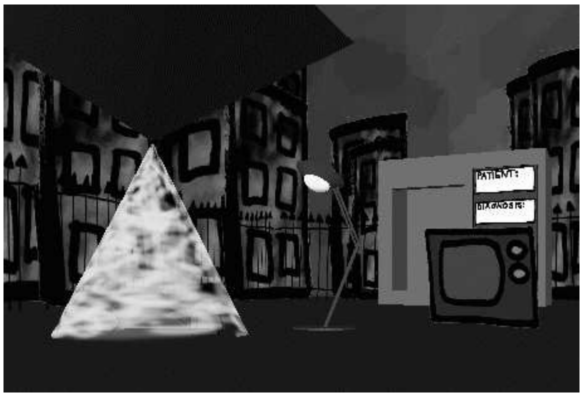


FIGURE 7.51: The End

is happening and that the machine is somehow involved. The Patient's glances at the camera seem to draw users into the scenario, generating a greater sense of connection with the scene and sympathy for the Patient's plight.

Transitions as Mindset Change

These case studies are only two examples of how transitions work; much more work needs to be done in order to explore how much of a difference they can make. But they do suggest that transitions change the qualitative perception of behavior by changing the nature of behavior from stimulus-response to reflection on the implications of what is happening around the agent. Transitions also change the way in which the designer tends to think, because they encourage the designer to think about and then make crystal-clear for the user the intended point of each behavioral change. This change in mindset ends up changing the nature of agent design in the Expressivator; we will explore these issues in the next section.

Agent Design in the Expressivator

Through its focus on transitions, the Expressivator changes the way designers must think about — and therefore go about — agent design. In Hap, the Expressivator's predecessor, an agent is defined in a number of steps:

1. Decide on the high-level behaviors in which the agent will engage.
2. Implement each high-level behavior, generally in terms of a number of low-level behaviors and some miscellaneous behavior to knit them together.
3. Use context-conditions, conflicts, and other design strategies to know when each behavior is appropriate for the creature to engage in.

The Expressivator more tightly constrains the agent design process. Similarly to Hap, the designer must first decide on a set of high-level signifiers the agent will express. But s/he must also decide on the transitions between the high-level signifiers; this includes deciding both which behaviors may lead to which other behaviors and the reasons the agent might want to make each behavioral switch. Similarly, for each high-level signifier, s/he must decompose it into a set of low-level signifiers and then explicitly decide how those low-level signifiers will interrelate.

Specifically, when building a high-level signifier, the agent designer must do the following:

1. Identify the low-level signifiers of which the high-level signifier is composed.
2. For each possible transition between low-level signifiers, determine the possible reasons for behavioral change (transition triggers).[3]
3. For each possible reason, determine how that reason should be communicated to the user (transition demons).

An example of such a design for the Patient is in Figure 7.52. Having made such a design, the intrepid agent builder must then implement each low-level signifier, transition trigger, and transition demon to create the high-level signifier.

Once the builder has engaged in this process for each high-level signifier, the high-level signifiers must be combined to form the complete agent. This involves a similar process of identifying each possible reason for each possible transition, and how each reason should be communicated. An example of this reasoning for transitions out of the "Turned Off" high-level signifier is in Figure 7.53.

In the end, the design of an agent involves two levels of atomization, with the atoms at each level interrelated through the use of transitions. The full structure of the Patient is shown in Figure 7.54.

A case study of the entire Patient design process is in Appendix C; this may interest humanists as well as technical readers.

[3] It turns out that quite a few of the theoretically possible transitions do not tend to make sense in practice. So this step is not quite as painful as one might expect.

High-level signifier: Mope by Fence
Low-level signifiers:

1. Look out at world
2. Sigh
3. Walk up and down fence

Relationships between low-level signifiers:

Behaviors	**Reason**	**How**
1→2	Life is bad! Wish I was out there!	Stop looking a moment Lost in reverie
1→3	Bored with spot. Get better position.	Look in the direction I am planning to walk. Focus on something there. Walk, keeping eye on spot.
2→1	I'm sad, but I still want to look	Interruption
3→1	Got to point where I can see the thing I want to look at	Turn to face and look at the thing intently

FIGURE 7.52: Design of the high-level signifier *Mope by Fence*.

New Behavior	**Reason**	**How**
Explore World Read Sign	Patient awakes from being turned off	Slowly rise up. Shake self. Blink, blink. Maybe sigh. Look around slowly to get orientation. This should be exagerrated the first time; after that it becomes a routine.
Exercise	Same reason	Here you should be exercising like a maniac while looking around for the Overseer. Taper off.
Mope by Fence	God, just another reason to be depressed	Same as transition to Explore World, but make it even more depressed.

FIGURE 7.53: The design for transitions out of Turned Off.

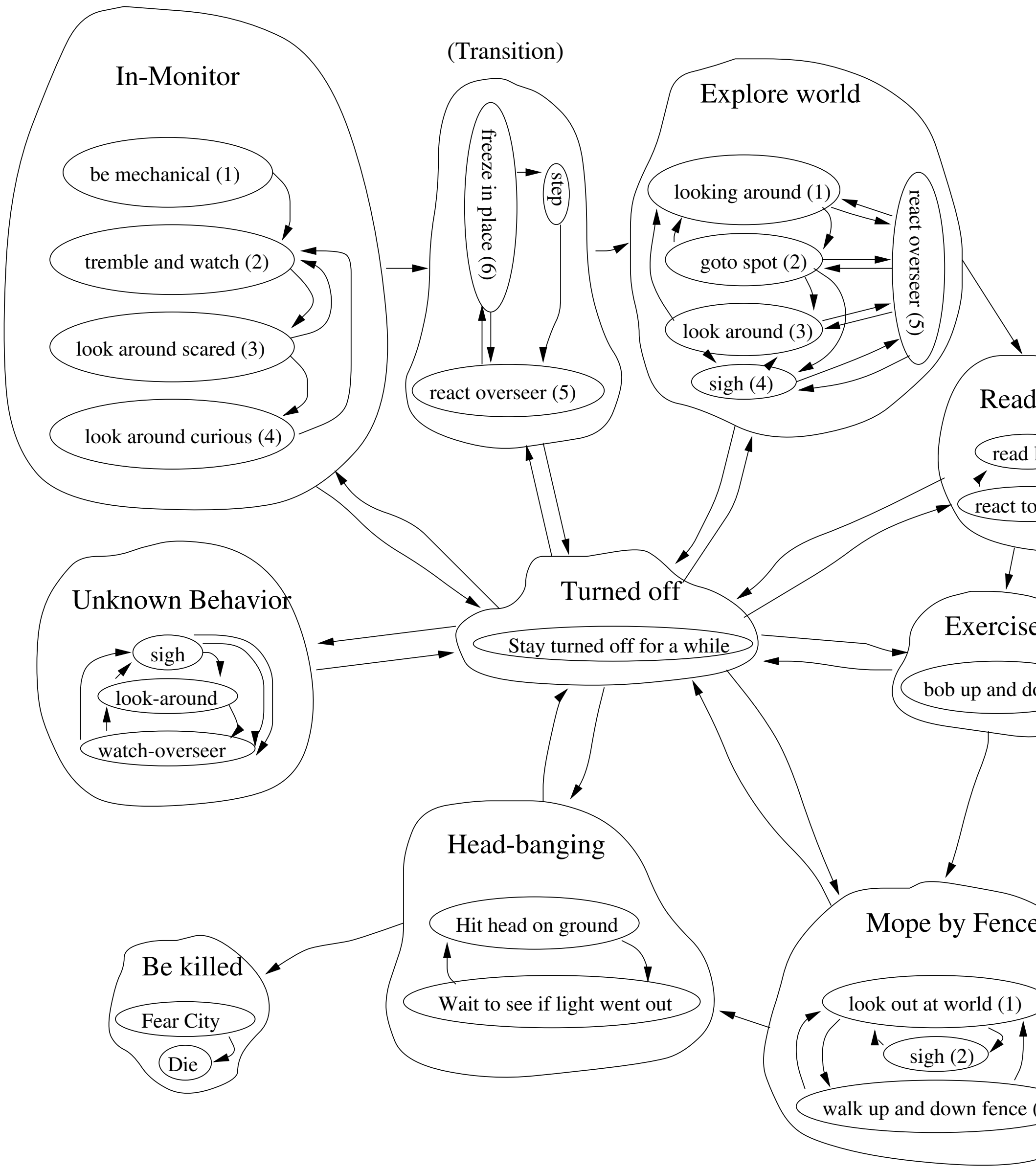


FIGURE 7.54: The complete design of the Patient, as implemented.

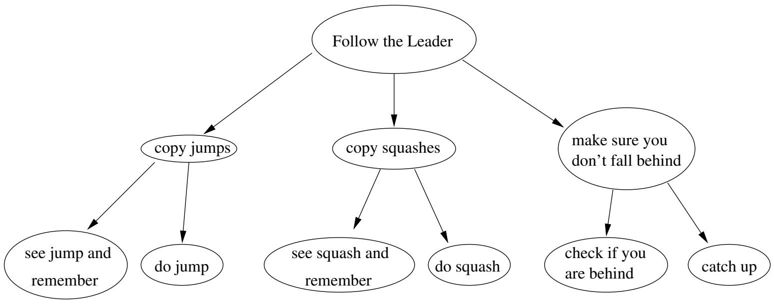


FIGURE 7.55: Follow the Leader structure in Hap

The Expressivator Mindset

In order to understand the Expressivator mindset more deeply, technical readers are suggested to take a moment now to read section D.2 of the Appendix.

The Expressivator changes the way the designer needs to think about behaviors. Because of the focus on transitions, the Expressivator demands that the designer know *why* the agent does what it does. In designing the Patient, I would many times want the agent to change behaviors, and discover to my surprise that I had no idea why the agent should change. I would be forced to stop and think about the reasons for the agent's behaviors; the articulation of those reasons would invariably clean up the behavior design.

But the change in mindset the Expressivator brings about goes deeper than this; it comes about from interactions between transitions and the redefinition of behaviors as signifiers. As discussed in Chapter 5, under the Expressivator framework behaviors are fundamentally things to be expressed to the audience. Complex behaviors may make an agent more intelligent, but if the audience cannot understand the complex nuances of the behavior, they are useless. Instead, under the Expressivator behaviors are simplified; the focus is on making them expressive. Instead of having complexity in the behaviors, *complexity comes from expressing to the user the interconnections between the behaviors*.

This change in mindset means behavioral code is structured differently. For example, when I worked on the Woggles, I built a behavior for following someone while playing the game Follow the Leader. The structure of this behavior is shown in Figure 7.55. The high-level behavior is broken up into three low-level behaviors, which all run simultaneously. Two behaviors are responsible for copying the leader's actions. One watches for the leader's "squash" actions, remembers how much the leader squashes, and squashes however the leader squashed last. The other watches for the leader's "jump" actions, remembers where the leader is jumping, and jumps wherever the leader went last. The third behavior is responsible for error recovery; it senses where the leader is, and if the leader is getting too far ahead, it takes over with a "catch-up" behavior that runs to where the leader is without bothering to copy the leader's actions.

The Follow the Leader behavior works well and robustly. The agent correctly follows what the leader does and is able to recover if the leader

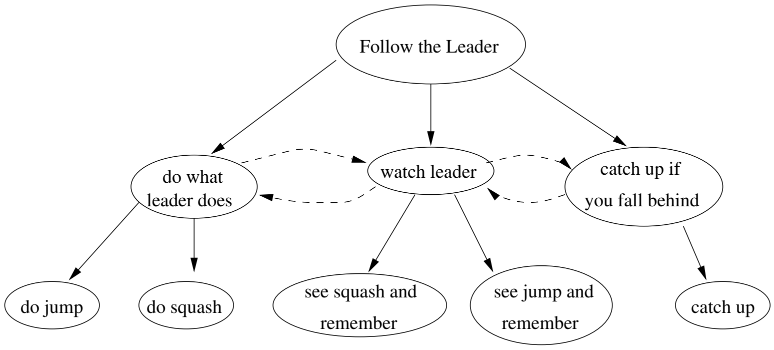


FIGURE 7.56: Follow the Leader structure in the Expressivator

From	To	Reason	How
Watch	Copy	Saw what the leader did	Turn glance from leader to where you are going
Copy	Watch	Want to know what leader does next	Pause; turn eyes to leader
Watch	Catch up	Can't see what leader is doing	Pause; strain to see leader; get nervous
Catch up	Watch	Caught up to leader	Pant; do subsequent behaviors more quickly

FIGURE 7.57: Transitions for Follow the Leader

is going too quickly for it. The flaw in it from the Expressivator's point of view is that the behavior is organized according to the logic of the activity, but *not according to what it is logical for the user to perceive*. We want to communicate to the user that the agent is watching the leader and copying its actions. But the actions of perception are split among all three behaviors and are generally done without corresponding movements of the agent's eyes; the action of copying is split into two completely independent behaviors. The Follow the Leader behavior is elegantly designed, but not optimal for communicating to the user what is going on.

Instead, a version of Follow the Leader for the Expressivator would require breaking up the activity of following into the things we would like the user to pick up on and their interrelationships. In Figure 7.56, we can see what such a structure might look like. Follow the Leader is now broken up into the behaviors that correspond what we want the user to notice: (1) watching the leader to find out what it is doing; (2) actually copying the leader's movements; and (3) catching up when the agent is behind. Each of these behaviors can be written relatively simply; the goal is not to do complex reasoning but to be sure to display clearly the basic idea of the behavior. Transitions (in dotted lines) are added to make the relationships between these behaviors clear; what these transitions mean is shown in Figure 7.57.

The heuristic of simplifying agent structure by focusing on its expressive aspects does not merely apply to the structure of the behaviors; it affects the entire design of the agent. For example, I built a rudimentary emotional system for the Patient (described in more detail in section A.2.5 of the Appendix). Originally, the Patient had a fear variable that would rise when the Overseer was near, and diminish when the Overseer went away. I then used the level of the fear variable to affect the Patient's behavior. The trouble with this system was that the Patient did not necessarily show any reaction to the Overseer's presence in conjunction with the change in fear. This means its fear would rise and fall without that fact being displayed to the user, making subsequent fearful behavior on the part of the Patient seem to come out of the blue. I therefore replaced this system with one where fear is increased whenever the Patient *visibly reacts* to the Overseer's presence. This model, where fear is the effect rather than the cause of fearful behavior, is psychologically dubious, but helps to ensure that users are kept apprised of the Patient's emotional situation.

Finally, it is not only the structure, but the content of behaviors that changes. Because the whole point of low-level signifiers is to communicate the agent's activity clearly to the user, most of the work in designing these signifiers is in working out the actual physical presentation of the behavior to the user. Rather than spending a lot of time on structuring code according to various conditions under which it should be engaged, the designer must spend substantial time with an animation package working on the details of motion. In some sense, the Expressivator reduces the problem of behavior generation to animation.

This emphasis on simple and extremely clear behavior contrasts with much current behavior-based AI work, in which the actual animated or robotic presentation of behaviors is considered trivial or beside the point. For the Expressivator, the level at which the basic units of meaning are communicated is essential; therefore, the graphical embodiment and manipulation of the agent, though perhaps not an "AI problem," is not a pleasant side-light but an essential part of what it means to be an agent.

Animation and Behavior-Based Programming: Battle of the Titans

The Expressivator demands that the agent designer spend substantial time getting the animated expression of the low-level signifiers right. I do not believe that I did this animation particularly well with the Industrial Graveyard. This is due to a number of reasons, starting with my continued subconscious inheritance of the AI concept that the code is the *real* agent and its graphical presentation only an afterthought. My lack of training as animator was another constant source of difficulty. But the major problem with creating adequate behavior for the Patient is that the substrate of the Expressivator, Hap, simply is not oriented to this way of thinking about agents.

I needed to use the Hap language in order to implement the low-level signifiers. Hap makes it easy (and fun!) to make complicated behavior with much variation based on the agent's mood, with reactions to any of a host of events that might be happening in the environment, with multiple processes running simultaneously, etc. What Hap does *not* do is make it easy to test and control low-level animation. This means it is relatively easy to build a dam-building behavior where a beaver searches for sticks and patches developing holes while keeping an eye out for predators; but it is relatively hard to create a sighing behavior that looks like sighing but not like panting or breathing. Getting the animation right in Hap is hard.

Humanists, don't worry if this paragraph is ununderstandable. Just move on to the next paragraph.

There are several difficulties with using Hap to generate animation. The most straightforward one is that Hap is a compiled language. That means in order to design a behavior one first writes a program, then one compiles it (a process that may take several minutes),[4] then one runs it. If the squashing looks just a touch too slow, one modifies the program, compiles it again, and runs it again. If now it is just a touch too fast, one goes through the whole procedure again. Every micro-change in the parameters of the code means several minutes of waiting before the designer can see the effect; the end effect is the designer feeling heartily encouraged not to fine-tune behavioral presentation.

But this problem is relatively easy to address. I did it by writing a Hap interpreter (which was itself written in Hap!). The interpreter would read in and execute new versions of the behavior while the simulation was running; low-level behaviors could now be tested and changed in the blink of an eye.

A more fundamental problem with Hap is the split it makes between the action architecture and the body. The gap between the actions that Hap produces and the actual movements the body ends up making as a result swallows up many fantasies of control of animated expression the agent designer may have.

Specifically, the agent's body is an articulated figure with 19 degrees of freedom, including such things as the body's position in space, the angle at which the agent holds its head, the body's color, and whether the body's light is on. As described in Chapter 5, rather than manipulating these parameters directly, Hap sends commands to the body at the level of actions; the motor system which receives these commands is then responsible for implementing the actions in a reasonable way given the physics of the world and other attributes of the body. For example, instead of telling the body to move to a particular point, Hap sends a command to "jump" to a particular point, reaching a particular height along its way. The motor system then calculates, based on the gravity of the world, what arc the jump should take and how much the agent should squash at the beginning and end of the jump. It also combines the physical manifestation of the jump correctly with actions that take place before and after the jump; so if the agent will continue into another jump, the motor system combines them so that the agent's momentum is carried through.

In this way, a single action generates numerous changes in the body's degrees of freedom over time in ways that depend on the agent's body, aspects of the environment, and the other actions that the agent has recently made or is about to make. This level of abstraction is essential because Hap is designed to control an agent in an uncertain environment, not generate pre-structured film clips. In essence, Hap sends down the

[4] To be precise, one compiles three times: once to turn the Hap into RAL code, once to turn the RAL into C code, and once to turn C into machine code.

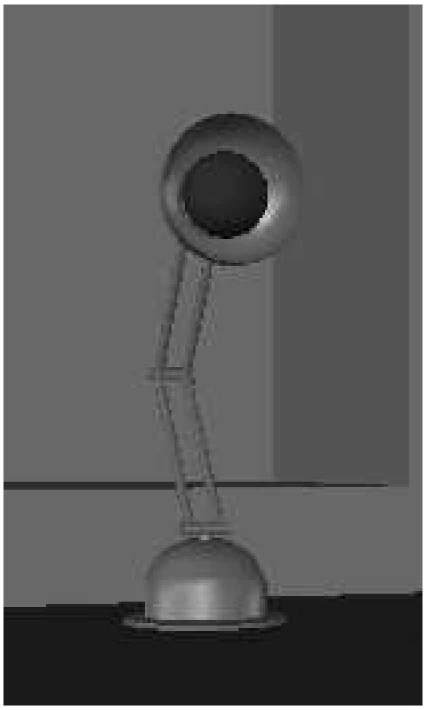


FIGURE 7.58: Headbanging movement 1

wishes of the agent's mind for what the agent will do, while the body fulfills these wishes as best it can given the constraints of the current situation, not all of which can be forethought by the designer or sensed rapidly enough by the agent. The motor system is needed because the run-time situation of the agent is uncertain; but it also means that *the action architecture* (and by extension the designer) *has no guarantee about the order or exact timing of the body's actions*.

But this exact timing is precisely what is at stake in generating expressive and clear animation. For example, when designing the Head-Banging behavior, I first used a keyframe editor to rough out the look of the behavior. Keyframe editors give direct, moment-by-moment control over the degrees of freedom of the body, immediately showing the effect of the chosen settings on the animation. Using the editor, it took about 5 minutes to generate a nice-looking animation.

The corresponding low-level signifier took days to implement. The behavior was not complex; the problem was not that the behavior would be incorrect. The problem was simply that you could not tell that the agent was purposefully hitting its head against the ground. Sometimes the lamp would look like it was flailing around; sometimes it would look like it was nodding; sometimes it would look like it was having a seizure; but rarely would it look like it was actually hitting its head on the floor. The difficulty is that head-banging involves multiple simultaneous actions: the agent must swing its head down while raising its body up (Figures 7.58-7.59), then swing its head up while bringing its body down (Figures 7.60-7.62), then snap its head back right before impact with the floor (Figures 7.63-7.64). All of these actions must be carefully timed,

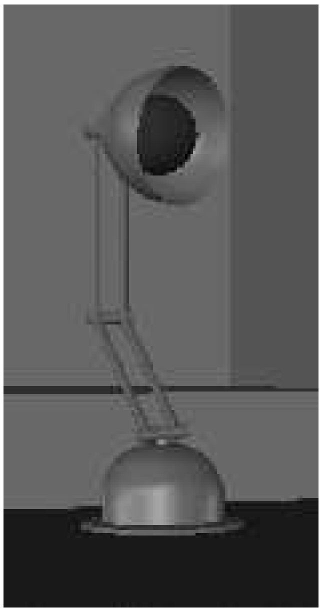


FIGURE 7.59: Headbanging movement 2

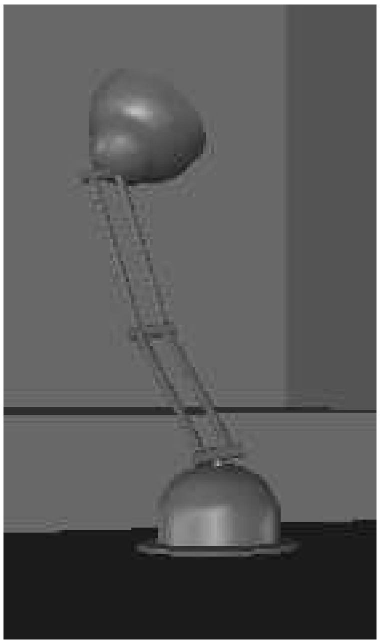


FIGURE 7.60: Headbanging movement 3

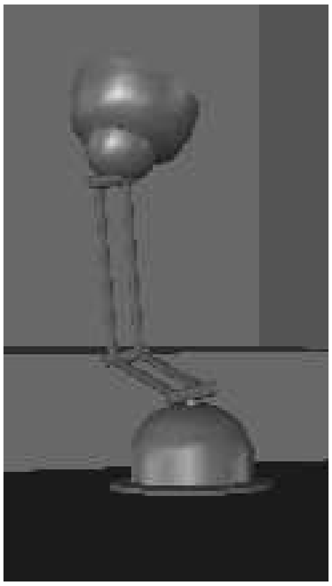


FIGURE 7.61: Headbanging movement 4

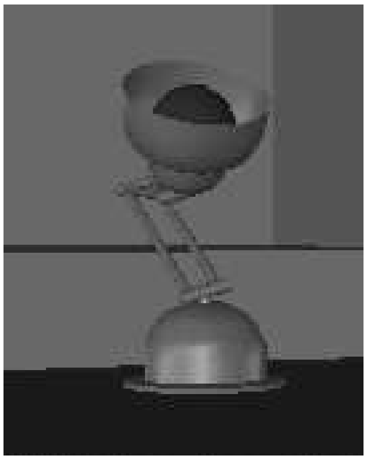


FIGURE 7.62: Headbanging movement 5

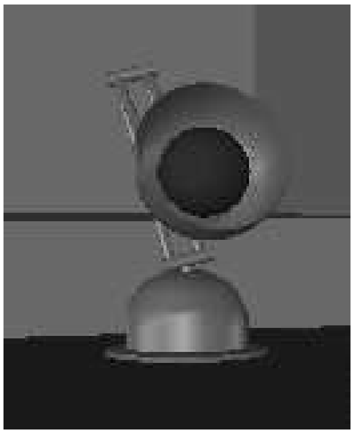


FIGURE 7.63: Headbanging movement 6

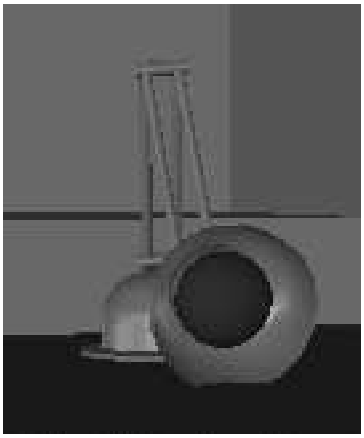


FIGURE 7.64: Headbanging movement 7

which is near to impossible in Hap; getting something that was remotely correct was a question of both luck and brute persistence.

The problem of generating expressive animation, while not a straightforward "AI problem," must be addressed by any architecture that is going to implement graphically presented, comprehensible agents. One promising avenue of exploration may be to use an automatic learning system such as genetic programming in order to generate code for the agent designer's desired low-level signifier. Automatic systems are easily able to generate many variations of behavior and test them rapidly in the virtual environment; these attributes could hopefully be harnessed to create the next generation of tools for expressive agents.

Expressivator: The Next Generation

If the problem of generating low-level signifiers is addressed, then the Expressivator suggests a new way of building agents. In the future, programming an agent might look like this:

1. Identify the agent's high-level signifiers.

2. Decompose the high-level signifiers into low-level signifiers.

3. Use machine learning to generate the low-level signifiers.

4. Identify mini-transitions between the low-level signifiers to make high-level signifiers.

5. Use machine learning to generate mini-transition sequences.

6. Write triggers for the mini-transitions.

7. Identify maxi-transitions between high-level signifiers.

8. Use machine learning to generate the maxi-transition sequences.

9. Write triggers for the maxi-transitions.

10. Tune everything by hand.

Transitions clearly add a new level of work for agent designers. Before, designers could content themselves to simply write behaviors. Now, designers must think about and implement many transitions between the behaviors. But in some sense transitions may actually *reduce* the complexity of the designer's job. Yes, you now need to write transitions, which was not necessary before; but transitions go between very simple behaviors with little internal structure, rather than the complex behaviors needed if one does not have transitions. And if you can generate most of the behavioral and transition sequences semi-automatically with machine learning techniques, in the end the behavior programming problem will be simplified.

Behavior Transition Types, Re-Visited

In Chapter 5, I argued for a range of transition types that the Expressivator should support. The Expressivator does, indeed, support all of the transition types I enumerated. Nevertheless, in practice I found that quite a few of the transition types were not useful. This is because the transition types are oriented towards blending or smoothing behaviors together. But for narratively expressive agents, the point is not to smooth behaviors but to make clear the relationships between them. Transition types that worked well to blur the distinction between behaviors worked poorly to explain the relationships between them; the reason for a behavioral change cannot be expressed when the user does not realize that the behaviors actually changed! Most of the mileage in transitions, then, comes from explanatory transitions; many of the other types were essentially clever tricks that do not help to make behavior more comprehensible.

Technical readers and curious humanists are now invited for a trip round section D.3 of the Appendix, which explains in more detail how each transition type was implemented, and whether or not it was useful.

Evaluation of the Expressivator

There are two aspects of the Expressivator that need to be evaluated:

1. *For designers*: Does the architecture give designers the controls that they need in order to implement the agents they may have in mind?

2. *For users*: Does the methodology behind the Expressivator actually create agents that are easier for users to understand?

Evaluation of the use of the Expressivator for the designer was part and parcel of the development of the Industrial Graveyard. In order to evaluate the effectiveness of the Expressivator in terms of what the user comes to understand, it would be best to do some kind of qualitative or quantitative user study. Unfortunately, this turned out to be beyond the scope of the thesis. In this section, I'll first explain the pluses and minuses of using the Expressivator to build an agent, and then discuss the ins and outs of how architectures like this one can be evaluated.

Evaluation for Designers

Advantages of the Architecture

One of the major goals of the system was to make it easier for designers to coordinate multiple high-level behaviors. This was successfully achieved. There is no doubt in my mind that behaviors are much easier to coordinate in the Expressivator than in Hap. This was underscored by my attempts to build the Overseer in Hap. Although the Overseer's activity is extremely simple, with clear conditions under which each behavior is appropriate, I spent many days trying to manipulate various Hap attributes to get each behavior to be engaged in at the right time. I finally gave up and let the Overseer use the behavior-killing meta-level control to delete old behaviors that were no longer relevant; without this hack it was simply impossible to control the Overseer well.

There are a number of problems with coordinating behavior in Hap that the Expressivator addresses:

- *The implicitness of behavioral choice*: In Hap, the choice of what behavior to pick at any time depends on a host of factors, including environmental conditions, priority differences between various behaviors and subbehaviors, and conflicts between behaviors. This means that getting a particular behavior to be chosen in a particular situation is a matter of manipulating multiple aspects of the agent design, not all of which have effects that can be straightforwardly understood. In the Expressivator, the designer writes triggers to cause behavioral change directly; having a behavior happen in particular circumstances means writing a single trigger that causes exactly that behavioral change.

- *The re-eruption of dormant behaviors*: Under Hap, when one behavior is chosen over another, the no longer chosen behavior remains in the agent's behavioral repertoire but becomes dormant.

Later, when the more important behavior is finished, the old behavior becomes active again. This works fine if the new behavior was a short interruption. But what also happens frequently is that the new behavior runs for quite some time; after it is done, the agent leaps back to an old behavior that has lost all relevance.

The Expressivator deals with this by actually deleting old behaviors when a new one takes over, instead of leaving them lying around to rear their forgotten heads later. I found that most of the time, behavior that has been interrupted for a long time should be started again from the top, instead of starting from whatever point the agent stopped at 5 minutes ago. The Expressivator makes this the default; in cases where the behavior should only be interrupted and not deleted, the special 'interruption' transition can be used instead.

- *Invisible behavioral interruption*: The problem of out-of-date behaviors suddenly becoming activated is compounded by the fact that in Hap, dormant behaviors, when re-awakened, do not actually know that they have been interrupted! Because they do not know they have been interrupted, they control the body as though there had been no lengthy break in their behavior, which is clearly wrong.

  For example, when building the Overseer I wanted the 'patrol' behavior to end automatically if it had been interrupted for quite some time; otherwise, the Overseer would try to return to whatever arbitrary point it had been walking to whenever other behaviors relinquished control of its body. Nothing worked properly except the extraordinarily simple measure of using the meta-level controls to kill the patrol behavior when you were doing something else more important. In the Expressivator, this problem vanishes because behaviors are deleted when they are interrupted; transitions explicitly inform behaviors when they are or are not active.

An additional major advantage of the Expressivator is the ability to clean up before and after behaviors. When switching from behavior to behavior, you have an opportunity to say something like, "I'm not reacting to the Overseer anymore, so I had better make sure to stop trembling and to squash a little less." For behaviors that have a large effect on body state — for example, that would involve the Patient tracking the Overseer, crouching down or stretching up, leaning over, keeping its eyes shut, or trembling — this opportunity to set the body back to a more appropriate state for the next behavior in some plausible manner is invaluable. Without it, the Patient has a good chance of repeating some of the major Woggle bugs: trembling or having its eyes shut through multiple subsequent behaviors, until some behavior serendipitously resets the body state.

But transitions do not seem essential to doing this clean-up activity. One possible way of doing this without transitions is to have a generic clean-up behavior, which you call before you start any behavior. I tried this with the Patient, but generally speaking this gave the look and feel of resetting the body after each behavior to a known state (the equivalent of Silas's standing up between behaviors mentioned in Chapter 5), which did not look good. Instead, I just made sure the transitionless version of the Patient avoided the most egregious behavioral carry-over by stopping trembling before every behavior. Nevertheless, there are still frequently problems in the transition-less version with inappropriate body aspects from previous behaviors carrying on into the next one.

A nice approach in general might be to define a clean-up behavior for each behavior. This clean-up behavior would reset aspects of the body state that the old behavior manipulates and that would probably be wrong in subsequent behavior. With the transition system, you know when a behavior is ending, so you could automatically call the clean-up behavior whenever the behavior was about to be deleted. This generic clean-up could occur in addition to whatever specific body changes were necessary for the next behavior.

Problems in the Architecture

There were certainly problems in the architecture. Of these, the most egregious is the problem of generating adequate animation. as discussed above. There were also some technical difficulties with the use of Hap as the basis language for the Expressivator, the most important of which is described in section D.4 of the Appendix for the benefit of technical readers; now would be a good time to take a look at it.

The major difficulty I ran into with the Expressivator per se (not its Hap substrate) is in reactivity. Specifically, in Hap, when you switch behaviors, the old behavior simply becomes dormant. The Expressivator, on the other hand, actually needs to delete the old behavior, including all its subbehaviors. This tended to add unwanted overhead to the time to switch — not much, perhaps 100 milliseconds, but enough to be noticeable in a delayed reaction time. One possible solution to this would be to simply mark behaviors as deleted, rather than actually deleting them; the agent could go back and actually do the work of deletion when it has more time to think.

Conceptually, though, the greatest problem with the Expressivator is the potential explosion of the number of transitions needed between signifiers. With 5 signifiers, there are up to 25 possible transitions; if an agent has 100 signifiers, there are far too many transitions to write by hand. From this perspective, the Patient has 24 signifiers, so it seems superficially like it would require just under 600 transitions!

But there are a number of factors, some theoretical, some practical, which cut down greatly on the number of actual transitions needed. An important factor in cutting down the number of transitions is the split between low- and high-level signifiers. Transitions are only needed between high-level signifiers, and between low-level signifiers that share the same high-level signifier — *not* between low-level signifiers in different high-level signifiers. This means that the Patient, with 8 high-level signifiers and 15 low-level signifiers grouped in small clusters, would require at most just under 150 transitions (64 maxi-transitions and 82 mini-transitions). In general, if we assume that low-level signifiers are distributed more or less evenly among high-level signifiers (rather than, say, all being under the same signifier), this reduces the original $O(n^2)$ problem to one of $O(n\sqrt{n}))$.

This number is still far more than I actually implemented. I reduced the number of transitions needed using several techniques. Interrupt transitions do 'double duty' by taking care of the transitions both into and out of a behavior. I cut out many transitions by writing several generic transitions, that could go from any behavior to a particular behavior.

Most importantly, I found in practice that many of the possible transitions did not make sense because of the semantics of the behaviors involved. For the Patient's 8 high-level signifiers there were only 15 maxi-transitions, and for the Patient's 16 low-level signifiers, there were only 25 mini-transitions (this number could have been cut down even more if I had shared mini-transitions between the same low-level signifiers when used in different high-level signifiers). Granted, the Patient is not as complex as it could be; but even in the fully complex unimplemented design of the Patient (shown in Figure C.20), there were 27 maxi-transitions, meaning under half of the possible maxi-transitions actually made sense.

One way to address this problem even further is to use generic transitions for most cases, and specializing them when the generic version is inadequate. For example, the transition out of sigh is always the same, unless sigh is returning to looking around. In this case, going directly from sighing to looking around the world looked odd, since the sigh was very slow and looking tends to consist of quick glances. Therefore, I made a generic sigh transition, then specialized it when going to looking around by adding a slow look. This slow look mitigated between the slowness of sighing and the speed of looking. This is one way to cut down on the complexity of number of transitions; make general ones for everyday use and add small touches for specific cases.

Finally, the separation between the motor system and the action architecture which causes such problems with animation also undermined the agent's ability to physically connect behaviors. When moving from one behavior to another, the agent needs to be able to sense accurately where the body is in order to be able to engage in a proper action sequence leading to the next behavior. The difficulty with the motor system / action architecture split is that you can sense where your body is, but you don't know where it will be when whatever acts that are currently being executed by the motor system are finished. This problem would probably need to be addressed by being able to get more information from the motor system about the position in which the agent can expect the body to be before whatever actions it is currently taking will be scheduled.

Evaluation for Users

Ease of use for the designer only answers some of the questions raised by this thesis. Given that the designer is satisfied with the created agent, that does not yet mean that users will interpret agents in the way the designer intended. Several possible questions still arise:

1. Do users recognize the behaviors the designer is trying to communicate?

2. Do users understand the connections between behaviors that the designer has in mind?

3. Does the addition of narrative sequencing really make the agent seem more intentional?

The detailed analysis of two transitions earlier in this chapter certainly suggests that, with the Expressivator, the user is given more information on which to judge both the agent's behavior and the reasons for the agent's behavioral changes. This is certainly a basis for improved user understanding, but does not necessarily imply actual improvement. In particular, the quality of the animated behavior is not up to snuff, which means users sometimes have trouble interpreting the simple movements of the agent; the animated presentation of the Patient would have to be fine-tuned in order to make the differences in comprehensibility truly striking. Anecdotal evidence from demonstrating the system suggests that the agent appears more intentional or 'alive' with transitions, but the system has hardly been tested under rigorous enough conditions in order to definitively answer these questions.

One reason this testing has not yet been done is because the goal of agent as communication (rather than as a functional tool) problematizes the question of evaluation. A respected technique for testing systems' desired effects on users is to do statistical studies of the impact of the system on various users. One can then conclude that the system is effective if there is a statistically meaningful effect across the pool of users.

But this adequacy across users is not necessarily the best technique to use when the goal is communication. For example, suppose that the agent is in practice incomprehensible to many users. But for a small subgroup of the target population, the agent is not only comprehensible, but makes an enormous and lasting impact on the way in which the users think and lead their lives. For some agent designers, this result may be much more satisfying than an agent which has a marginal impact on many users. Basically, statistical tests may be inadequate for such designers to evaluate the quality of their agents for the same reason that best-seller lists are not necessarily the best technique to judge the quality of a novel: a deep impact on a few people may be much more valuable than a shallow impact on many people. Issues such as this one will have to be explored by researchers delving into this area before we can be confident that the tests we are using are truly meaningful.

But as a first pass, I propose the following technique for evaluating a system like the Expressivator rigorously. Users interact with one of two versions of the system: one with behavior transitions and one without. Users are videotaped using the system, while talking aloud about (1) what they think the agent is doing and (2) why they think the agent is doing it. These protocols can be compared with the designer's intended behavioral communication at each step. Analysis of these videotapes is necessarily subjective (though not arbitrary), since there is no way to determine a meaningful 'quantitative distance' of the user's verbal interpretations from the designer's perhaps not entirely articulated intentions for the system.

If there is a need to get more quantifiable results, users could be surveyed after the video session using statistical techniques similar to those of Scott Neal Reilly [Neal Reilly, 1996] or James Lester [Lester *et al.*, 1997]. They could be asked, for instance, about their perceptions of the agent's personality; presumably, if they understand the agent's behavior and motivations, they will end up with a better understanding of the agent's personality over all. Some open-ended questions on the questionnaire, modeled on Lester's "who does Herman the Bug remind you of?" could round out study of the user's understanding of the agent.

The Expressivator as Narrative Intelligence

The Expressivator is intended as one example of what Narrative Intelligence might look like. The most obvious instantiation of narrative principles in the Expressivator is the use of transitions to form narrative sequences from atomic behaviors. But the narrativity of the Expressivator is more complex, involving not only the technology of the system — signs, signifiers, and transitions — but also such aspects as the philosophy and context of the Expressivator's use that normally do not count as part of the system, technically conceived. This makes sense, since narrative is, in the words of Katherine Hayles, *emergent*: it is a property not of artifacts conceived in isolation, but of those artifacts in the contexts in which they are used and interpreted [Hayles, 1997]. Here, I will review each of the properties of narrative and explain how it is embodied in the use of the Expressivator:

- *Narrative Diachronicity*: Narratives focus on events as they occur over time; similarly, the Expressivator's transitions relate the agent's activities to one another.

- *Particularity*: Narratives are particular; they are not just about abstract concepts, but about particular details. In using the Expressivator, the actual details of animation by which the agent's behaviors are communicated to the audience are similarly essential. Many behavior-based systems leave out this articulation of behavior into its physical presentation, but when a graphical system is intended to communicate, those behaviors must be specified down into the details of movement with a particular body.

- *Intentional State Entailment*: When interpreting a narrative, people focus not so much on what the agent is doing, but on how it feels about what it is doing. Transitions function here to regularly communicate what the agent is thinking about its actions: not just what it does, but *why* it does it.

- *Hermeneutic Composability*: Hermeneutic composability refers to the complicated interrelationships between the interpretation of various pieces of the system. One cannot focus simply on each particular component in isolation, but must look at how they interrelate.

  In the Expressivator, the agent's behaviors do not exist in a vacuum; using transitions, they are brought in relation to one another, both at the level of presentation and at the level of design. The agent's actions become signs and signifiers in ways that depend on the context of interpretation; moving the agent's head could result in a 'nodding' sign in one place, and a 'shaking to get light on' sign in another. These different signs cascade into different signifiers, meaning the agent's understanding of its actions is context-sensitive in ways that approximate those of the hermeneutically reading user.

- *Canonicity and Breach*: Someone's behavior will appear narratively comprehensible when it involves a set of expectations which are set up and then violated; the person must not be entirely predictable. Similarly, the Patient is set up so that there is much variation in behavior; every time the Patient hits its head, for example, it chooses slightly different angles and speeds of attack. The plot itself is an excellent example of story based on canonicity and breach: the Patient is turned off, turned off again, turned off again, until the last time, when instead of the expected turning off, it is killed.

- *Genericness*: The genre expectations of the user, which form the basis for understanding what the system is about, are set up in the context of the system. This means that the proper use of the Expressivator does not limit itself to the construction of behaviors and transitions within the agent. Rather, the Expressivator focuses on the likely user interpretation of the agent, which itself may be influenced by a host of contextual factors. In the Industrial Graveyard, correct interpretation by the user is set up, not only through the Patient's behaviors, but through the design of the user interface (e.g. the graph showing how good or bad the Patient is being), through the informational brochure which users read before they begin to interact with the system, and through the decoration and lighting of the virtual environment. These factors are not external to the system, though they are external to the technology of the Expressivator; they set up the context within which the Patient's behavior will be interpreted.

- *Referentiality*: In a story, the 'facts' are not paramount. Similarly, in the Expressivator, the agent's behaviors are not an absolute, which is then to be communicated as an afterthought to the user. Rather, the agent's behaviors are oriented to and dependent on the interpretation of the user. In this sense, the agent is a narrative, rather than a pre-existing problem-solver.

- *Normativeness*: Narratives depend on the audience's conventional expectation about how people will act. These expectations are here used as a basis for behavioral design. I designed the agent's behavioral sequences on the basis of background knowledge of how the audience would likely interpret the agent's behavior. Nevertheless, the overall experience could have been enhanced by more carefully thinking out the nature of the target audience. My general assumption was that the piece is oriented towards people who think like me.[5] Exploration of ways to explicitly tailor agent presentation towards particular audiences is an essential component of future work.

[5] In that respect, I am no different from many other AI researchers — the only difference is that I explicitly recognize that I am making an inaccurate assumption!

- *Context sensitivity and negotiability*: In Chapter 6, I say that serving up prepackaged narrative without leeway for audience interpretation is throwing away the best properties of narrative. Nevertheless, that is exactly what I do here. I decide on all the behaviors and transitions ahead of time, and then the goal is simply to make sure that those decisions make it across the yawning divide to the user intact.

  In this respect, signs can be taken too literally. If signs are thought of as absolutely everything that must be communicated, one by one, to the user, we end up merely replacing behavioral atomization with signifying atomization. An agent that is so simply and straightforwardly understood is too easy.

  A very different approach that is much more friendly to the value of negotiability is that taken by Simon Penny in his robot, Petit Mal [Penny, 1997a]. The design of Petit Mal explores the extent to which people can attribute meaningful behavior to autonomous robots. Petit Mal is set up, not to elicit any particular behavioral interpretation, but to allow for many possible behavioral interpretations. Far from trying to impose particular interpretations on the user, Penny uses Petit Mal as a blank screen onto which many possible interpretations can be projected. Petit Mal is interpretationally plastic, and never exhausted by the onlooker's musings; this gives its dynamics a degree of liveliness which the Patient lacks.

  The difficulty with this plasticity is that it is relatively low-level. At the internal level, Petit Mal does some simple navigation and obstacle avoidance (which is, of course, regularly interpreted as much more complex behavior). It is not clear how much more complex behaviors can be constructed for Petit Mal without simultaneously greatly constraining the interpretational space. In this sense, Petit Mal and the Patient occupy more or less opposite ends of the spectrum of interpretational negotiability on one end and understandable complexity on the other. If this is so, it might be interesting to now try working for something in the middle.

- *Narrative accrual*: It is not clear how narrative accrual would apply to the work I have done here.

Fundamentally, narrative is more about the *quality* of behavior, rather than its *correctness*. Because this attitude differs from that of the action-selection approach at the heart of behavior-based architectures, a number of changes to the behavior-based framework are necessary. Fundamentally, behaviors should be simple and expressive; intentionality is communicated to the user by clearly displaying the relationships between behaviors. The detailed technical changes the Expressivator makes to Hap are summarized in section A.4 of the Appendix. In general, the changes the Expressivator makes can be summarized as follows:

- Instead of breaking behaviors into physical actions and behaviors, the Expressivator breaks behaviors into signs and signifiers that are communicated to the user. The agent keeps track of the user's likely current interpretation through the sign-management system, which posts signs and signifiers once they have been expressed, allowing the user's likely interpretation of agent activity to influence the agent's behavioral decisions.

- Instead of simply atomizing the agent's activity, the Expressivator includes transitions that express to the user the agent's reasons for changing from one behavior to another, simplifying the user's comprehension of the agent as narrative.

- Instead of having behaviors being basically independent, the Expressivator gives them meta-level controls by which they can coordinate with one another to give the user a coherent picture of the agent's personality and intentions.

The Expressivator combines these systems to try to allow designers to build agents that express their activities and thinking to the user, without giving up many of the advantages that behavior-based architectures can provide. The as yet untested hope is that these agents will appear, not only more understandable, but also more visibly alive.

Chapter 8

Conclusion

In this thesis, I have taken you on a long and circuitous intellectual journey, and now, at the end, it is time to go back to the beginning and see how it all fits together. From a computer science perspective, we tackled problems in integration for behavior-based autonomous agents. We found some inherent limitations in the ability of standard AI methodology to ever fully integrate agents, but discovered ways to mitigate the effect of this underintegration by redefining agents as channels through which agent designers communicate to their audiences. This changes the focus in agent-building from primarily a design of the agent alone, with its communication as an afterthought, to including the agent's comprehensibility in the design and construction of agents from the start. This rethinking of the nature of agents led to the proposal that if agents are to be comprehensible as intentional beings, they should be structured to provide cues for narrative interpretation, the manner in which narrative psychologists have found people come to understand specifically intentional behavior. The Expressivator was developed as one architecture for this 'Narrative Intelligence.' It combines (1) redefinition of behaviors as signifiers and their reorganization in terms of audience interpretation, (2) the use of transitions to structure user-recognized behaviors into narrative sequences, and (3) the use of meta-level controls to strategically undermine over-atomization of the agent's behaviors. Preliminary results are encouraging, but further work, preferably involving the development of support for graphical presentation, will be necessary in order to fully evaluate the implications of and possibilities for the architecture.

From a cultural theory point of view, we started with the identification of a technical problem in computer science with remarkable similarities to some notions of schizophrenia in cultural theory. These similarities are not a coincidence; rather, they can be traced to atomizing methodologies AI inherits from its roots in industrial culture. The disintegration AI researchers can recognize in their agents, like that felt by the assembly line worker and institutionalized mental patient, is at least in part a result of reducing subjective experience to objective atoms, each taken out of context and therefore out of relationship to one another and to the context of research itself. This suggests that the problems of schizophrenia can be mitigated by putting the agent back into its sociocultural context, understanding its behavior as implicated in a cycle of human interpretation, on the part of both its builder and those who interact with and judge it.

Because this is a change in the metaphor at the heart of current systems, the embodiment of this changed perspective in technology has implications beyond some technical tweaks on a pre-existing and unchanging technical base. Instead, it changes at a fundamental level the meaning and usage of many parts of the system, even those that were not intended to be affected, and suggests even in its presentational failures the communicational limitations of a system which sees the essence of an agent purely in terms of the abstractions of its internal code.

This sums up the computer science perspective on the one hand and the cultural theory perspective on the other. But in the introduction, I told you this thesis is not half computer science and half cultural theory; rather, it is a single body of work which can be seen in various ways from either perspective. Now that you have had a chance to see both sides of the coin, I will take some time to step back from the details and discuss the implications of this work from a combined perspective, the one I developed, at times against my will, during the work this thesis represents. First, I will return to the notion of narrative and summarize its relations to schizophrenia and atomization. Then, I will step back to the meta-level to review the role of this thesis as a subjective technology, as one way to synthesize humanistic and engineering perspectives into a knowledge tradition that bridges the enormous chasm splitting contemporary Western intellectual life.

Narrative and Schizophrenia

This section is speculative; it is not intended to represent any kind of final truth. Instead, I will connect up the various strands of the thesis, to form a picture of where they seem to lead as a whole.

In this thesis, we started with schizophrenic agents, and ended up with Narrative Intelligence. Narrative became important for agents when it became clear that default technical approaches to hiding atomization from the user were not helpful in making agents seem intentional. In order to understand intentional behavior, users attempt to construct narrative explanations of what the presumed intentional being is doing; but this approach conflicts with the mechanistic explanations designers themselves need to use in order to identify, structure, and replicate behavior.

This contrast between narrative explanations that explore the meaning of living activity and atomistic explanations that allow for the understanding and construction of mechanical artifacts repeats the criticisms of anti-psychiatry. R.D. Laing and other anti-psychiatrists, after all, complain that the difficulty with institutional psychiatry is that it reduces the patient to a pile of data, thereby making a machine of a living person. Their solution — contextualization — seems at first blush to be a different response than the focus on narrative here. But just as we have seen that science is generally atomizing, we now can see that the methodology of contextualization contrasts with this atomization by being itself, too, a form of narrative. Anti-psychiatry follows the narrative tradition in the following ways: by structuring and relating the 'data' of a patient's life into the semi-coherent story of a meaningful, though painful, existence; by focusing on the patient not as an instance of a disease but as a particular individual and how that person feels about his or her life experience; and by relating the doctor's narrative to its background conditions and the life context in which it is created and understood. It is only through this process of narrative interpretation that anti-psychiatry feels the psychiatrist can fully respect and understand the patient's subjective experience as a human being.

If atomization involves thinking of human life mechanically, reducing it to a matter of cause-effect, while narrative allows for the full elucidation of meaningful intentional existence, then it seems likely that narrative — and by extension the humanities, for whom narrative is a modus operandi — can address meaningful human life in a way that an atomizing science simply cannot. If humans comprehend intentional behavior by structuring it into narrative, then AI must respect and address that way of knowing in order to create artifacts that stimulate interpretation as meaningful, living beings. This suggests that the schizophrenia we see in autonomous agents is the symptomatology of an overzealous commitment to atomistic science in AI, a commitment which is not necessarily unhelpful (since it forms the foundation for building mechanical artifacts), but needs to be balanced by an equal commitment to narrative as the wellspring of intentionality.

Humanists may recognize the arguments of Gadamer [Gadamer, 1986].

Schizophrenia in Postmodern Culture

But schizophrenia is not simply a difficulty of a contemporary agent-building method; schizophrenic subjectivity is also an important component of contemporary cultural theory. As discussed in the introduction, many cultural theorists identify schizophrenia as a way of thinking about contemporary human experience. This schizophrenia can be understood in a multitude of ways, but one way of understanding it is as a rejection of the idea that people are essentially unified, rational beings, with the suggestion instead that human consciousness is an emergent and somewhat illusionary phenomenon overlaying an actually fractured and distributed existence. While I am far from suggesting that we should go back to the idea that humans should be fundamentally rational, with emotion and meaning being mere distractions from the actual, logical, unified substrate of true humanity, my experience with schizophrenic subjectivity as it manifests itself in AI has led me to the conclusion that there are deep problems with the way schizophrenia is used in cultural theory, as well.

Note to technical readers: the rest of this section may be difficult for you to follow, as it uses the results from analyzing schizophrenia in agents to address a technical debate within cultural theory. I think you may find this different perspective interesting; but if you are feeling unhappy, please feel free to go directly on to the next section.

Specifically, schizophrenia comes about in AI when a living being's activity is reduced to simple atoms with limited interaction. Schizophrenia is in this sense the limit-point of formalization, the point at which important aspects of flowing existence are simply left out of the picture and therefore only appear as gaps or fissures between simply defined atoms. But this suggests that, in some sense, the postmodern (a.k.a. schizophrenic) subject, too, may be simply internalizing and celebrating the atomized view of itself that bureaucracy, industrialization, and modern science and technology have developed.

"The play of significations, their proliferation, their being out of gear with representations, because of the autonomy and arbitrariness of the way the stock of signifiers operates, has contradictory consequences: it opens possibilities for creativity, but it also produces a subject cut off from all direct access to reality, a subject imprisoned in a signifying ghetto." ([Guattari, 1984], 92)

The notion that this postmodern subjectivity is in some sense inherited from the technology we use is gradually becoming commonplace.[1] The idea here is simply an extension to this: that through the structure of the technology which is deeply interfaced with our daily lives, we imbibe the atomistic, objectivizing view of both ourselves and the interactive moment that that technology presupposes. The hyperbole

[1] For elegant descriptions of how this works in practice, see e.g. [Hayles, 1993] [de Mul, 1997].

surrounding hypertext is a case in point; its inheritances from scientistic self-understanding can be seen in its uncritical enthusiasm for technical development and frequent dismissal of criticism of that enthusiasm as neo-Luddism; in the notion of the 'postmodern narrative' as chunks of data with no overarching meaning, and only local structure; in the rhetoric of authorlessness, as though the text sprang from no context and was entirely ahuman; and in the movement of the responsibility for generating narrative understanding squarely onto the shoulders of hapless readers, who are left desperately trying to fabricate a narrative from randomly strewn atoms simply because they are good at it and hypertext technology is not.

If schizophrenia is something we are catching from our technology, then we must simultaneously ask ourselves if that is something we would like to catch. Though schizophrenia has multiple uses and I by no means intend to criticize all of them, I still have deep fears about the sometimes uncritical and whole-hearted postmodern importation of schizophrenia from modern technology as a new — and by extension positive — way of being. This is because the postmodern worldview is dangerously close to making the assumption that the ideas we import from technology come from some shining stratosphere of newness; rather than, as analysis of scientific work frequently makes clear, from a continuous cycling and recycling of metaphors and concepts from broader culture to scientific culture and back again. In the case of schizophrenia, these concepts are recycled from an industrial and institutional culture that most postmoderns would not knowingly choose to embrace, and that in fact only get their alluring aura by coming attached to our new high-tech toys. As Bruno Latour says,

> It is strange to say, but I think much of postmodernism is scientistic. Of course they no longer believe in the promises of science — they leave that to the moderns — but they do something worse: they believe in the ahuman character of science, and still more of technology. For them, technology is completely out of the old humanity; and as for science, it is almost extraterrestrial. Of course, they do not see that state of affairs as bad. They are not indignant at the ahuman dimension of technology — again they leave indignation to the moderns — no, they like it. They relish its completely naked, sleek, ahuman aspect.... I think that this is deeply reactionary because in the end, you push forward the idea that science and technology are something extraordinary, completely foreign to human history and to anthropology. ([Crawford, 1993], 254)

The antidote to this is, again, narrative: putting into context, creating origin stories about, attributing authorship, constructing meaning. This means in particular narrative to connect science and technology to the rest of our cultural life, reminding ourselves once again that science is done by people, that the views, strategies, and goals of those people is shaped, in part, by the culture in which they live. There is no law that says science must be atomizing, and that, by extension, technology must be schizophrenizing. Instead, we can return to the notion of subjective technologies, finding a middle ground between narrative and atomization.

Subjective Technologies

When we began, I set the goal of developing a kind of technology that respects and addresses the complexities of subjective human experience. This is in contrast to much current AI practice — many practitioners of AI, particularly those of the alternativist persuasion, are reluctant to engage in questions of what it *feels* like to be alive in the world. Subjective experience is often felt to be fundamentally illusory and unreliable, something to be replaced at the earliest possible moment with a more objective and testable form of knowledge. Building technology, in this way of thinking, may require a commitment to objectivity, since fuzzy mentalist concepts simply cannot be directly implemented.

In the work presented here, by contrast, subjective experience is essential — that is, the subjective experience of those who build and who come to interact with the agent. The mechanicity of current agents is a subjective experience, which can be fixed not by trying to find ways to make the agent objectively intentional (perhaps a contradiction in terms), but by respecting the subjectivity of that experience in order to enable it to be the best experience possible. The goals the designer has for the agent, independently of its actual effect, are, as well, a subjective factor — probably not completely definable, but nevertheless hopefully achievable through particular design strategies. In this sense, subjective experience and technology are by no means incompatible.

The work I have done here combines technology and subjectivity by seeing an agent as a form of communication, in terms of the intentions of its designer and how it is experienced by the audience. In this light, the major question to be answered is not "how can we objectively and testably reproduce experience?" but "what are the *goals* of the agent-builder in terms of how his or her agent design should be understood, and how can they best be fulfilled?"

The major change this philosophical distinction makes at the technical level is that *comprehensibility is seen as an essential requirement to be engineered in from the start*. Certainly, other AI researchers have been interested in making comprehensibility a goal. But, generally speaking, these attempts have come as an afterthought, at the point where the target user population expresses reluctance to interact with or trust intelligent systems whose behavior they do not understand. The field of expert systems, for example, has had a rash of mostly unsuccessful attempts to modify systems that make correct but obscure conclusions in order to make clear to human users how they came to them. My experience with the Expressivator suggests that it is so difficult to make already-designed systems comprehensible after the fact simply because comprehensibility cannot be adequately addressed through a set of tweaks added at the end. Rather, it requires changes in the way we structure and design agents from the beginning.

Anti-Boxology Re-Visited

This simple fact — that systems designed separately from the context of their human use may not function as well as ones that keep that context in mind from the beginning — brings us back to the postulates of anti-boxology I set forth in the introduction. The anti-boxological perspective sees life as inadequately understood when carved into separate categories; instead, it seeks to relate those categories to each other. When I introduced this concept, I stated rather mysteriously that this thesis would be anti-boxological on several levels: disciplinary, methodological, and technical. We are now ready to go back and look at the thesis as a whole as an instance of anti-boxological thinking.

At the disciplinary level, the engineering approach used here stems from and is continuously informed by a humanistic perspective on agent-building. Engineering and the humanities are not seen as two separate activities with little to say to each other. Instead, they are thought of as two (sometimes vastly) distinct perspectives, which can be profitably put in relationship with one another.

At the methodological level, the development of socially situated AI puts the agent into a sociocultural context that includes the people who build it and the people who observe it. This is reminiscent of the viewpoint of Terry Winograd and Fernando Flores, who argue that rather than thinking about how humans can communicate with computers, we ought to be thinking about how computers can enable better communication between people [Winograd and Flores, 1986]. Here, though, this does not involve the whole-sale rejection of AI, but a change in one of its fundamental metaphors. Instead of seeing agents as autonomous, socially situated AI argues that the agent should often be thought of as a kind of communication. In this agent-as-communication metaphor, the social environment of the agent is, not some unfortunate baggage to be discarded or ignored, but essential to and constitutive of the design of the agent. This change in methodology is directly represented in the technology through the shift in structuring agents from internally-defined behaviors to externally-observable and communicated signifiers.

At the technical level, the parts of the agent are explicitly put in the context of each other and of the agent's overall personality through the use of transitions. Transitions represent for the designer, and express to the user, the relationship between the different pieces of the agent. Meta-level controls provide the technical basis for interrelating behaviors in this way by allowing behaviors to coordinate to present a coherent picture of the agent's overall activity to the user. The details of the agent architecture therefore repeat the themes of the highest level of motivation: we have anti-boxology all the way through.

Lingering Questions

So far, I have discussed the way the thesis works at a high level and in terms of the themes I developed in the introduction. At this point the reader may have followed the argument, understood where we went and how we got there, but still have lingering high-level questions about the thesis. Here, I will try to answer some of the major questions that this work frequently brings up.

Questions from a Technical Perspective

> *As far as I can tell, the Expressivator adds some tweaks to an already-existing architecture in order to let the designer*

> *manipulate the audience's perception of the agent. Your agent doesn't actually become any smarter; the transitions all have to be written by hand. In what sense is this an AI contribution?*

It is true that this thesis follows in the tradition of much of behavior-based AI by designing behaviors — including transition behaviors — by hand. The agent's reasoning is minimal, compared to what some classical AI programs do. Like many other behavior-based systems, the agent makes behavioral decisions based on perception of the environment and memory of its own activities — although unlike these systems, it can also make decisions based on the likely user perception of its activities and based on tokens which represent the reasons for its behavioral changes.

The status of this design-oriented, direct programming approach to agents as a legitimate form of AI has been extensively defended by others (see e.g. [Agre, 1997]), and I will not repeat those arguments here. In addition to these general claims, the Expressivator has its own unique claim to being an AI contribution through its exploration of the changes that must come about in agent structure and design in order to allow agents to be comprehensible. Similar explorations have already occurred, most notably (but not exclusively) by Believable Agents researchers; this thesis adds to them by underlining the importance of *narrative* for human comprehension, and by outlining how this narrativity can be incorporated in AI, both in general in Chapter 6, and as specific technical mechanisms in Chapter 7.

This thesis does not simply provide some randomly-chosen technical hooks for user manipulation, but addresses the question, "what exactly is needed in order to make agents intentionally comprehensible?" It finds the answer in narrative: in order to be intentionally comprehensible, an agent must express not only what it does but also why it does it. The Expressivator then attempts to provide support for precisely this expression, by supporting the design and use of behaviors as communicated signifiers and by expressing the reasons for behavioral change through the use of transitions.

> *The Industrial Graveyard seems effective, but its effectiveness as communication are based on the use of conventions from animation, such as the exaggerated shock reaction to external events. How well do you expect this to map to other domains?*

The same conventions clearly will not work in radically different domains, such as photorealistic rendering. But clear communication is not simply a property of animation; it is also the goal of live-action film, novels, theater, and so forth. At its most fundamental, whatever the domain, the principles of narrativity still hold: the user still needs to be aware of what the agent is doing and why the agent is doing it. The difference between these domains is that *expression* of those activities and the reasons for them will need to be adapted to whatever domain the agent is built for, and however that agent is represented to the user. It seems likely that various kinds of autonomous agents will, over time, develop their own conventions of expressiveness, so that they will not need to be parasitic on more established genres.

> *I could barely wade my way through Chapter 3, but I still understand how the technology works. Couldn't you have built the technology without cultural studies, for example by simply importing suggestions from art and animation as you do in Intermezzo II?*

The short answer is yes, I probably could have — *but I most likely wouldn't have*. Once schizophrenia is identified as a problem, and once it is reframed in terms of agent communication, most of the technical answers I come to are straightforward. *The difficulty is in realizing that the problem needs to be reframed in the first place*.

The most important contribution cultural studies brought to the technical work, independently of any insights that I might have been able to glean purely from art, animation, and psychology, is the level of self-reflexivity that let me step back and realize that I was caught in a double bind: that atomization was both essential to code and the root cause of schizophrenia. Before I had this understanding I had already been trying to tackle the problem of schizophrenia for a number of years. Schizophrenia was at that time for me a gut feeling, not a well-defined concept, a feeling that there was something fundamentally wrong with the way agents were constructed, something that was inhibiting their intentionality. I came up with numerous technical proposals, many half-baked and some more complete, for addressing schizophrenia, each of which seemed upon reflection to repeat the very failures I was trying to address.[2] It wasn't until I realized, by comparing AI methodologies with the practices of assembly line construction and Taylorism, that what I was trying to do was simply and for good reasons not possible, that I realized I needed to rethink what I was trying to do in a deep way.[3]

The second most important contribution from cultural studies for the technical work came then, as I searched for a different way to think about agents that did not involve the same Catch-22: the suggestion on the basis of culturalist perspectives that the difficulty was that the agent is being taken out of context. Once I had the idea that the agent needs to be clearly communicated, much of the rest of the work could follow in a relatively normal technical way, using insights from various fields as they seemed appropriate (and in the manner to which AI researchers are accustomed). Nevertheless, for me the technical work is continuously informed, though perhaps in a less spectacular way, by my cultural studies perspective: from the understanding that interpretation is a complex process quite unlike simple perception to the ferreting out of the implications of changing the metaphors underlying agent architectures, this work is really cultural studies almost all the way through.

Questions from a Critical Perspective

> *Frankly, I find this 'AI Dream' of creatures that are truly alive to be ludicrous, if not downright Frankensteinian. In a world full of social problems, why should this goal matter to a cultural theorist?*

[2] This was a very trying time for my advisor.
[3] This was another trying time for my advisor.

The AI dream of mechanical creatures that are, in some sense, alive, can seem bizarre to those who are new to the idea. It is therefore important to note that this is not an idea that is new in AI, but, as Simon Penny notes, the continuation of a tradition of anthropomorphization that extends back thousands of years [Penny, 1995]. In this sense, the AI dream is similar to the 'writing dream' of characters that ring true, to the 'painting dream' of images that seem to step out of the canvas, to the fantasies of children that their teddy bears are alive, and to many other Pygmalionesque dreams of human creations that begin to lead their own lives.

But there is certainly a sense in which AI brings a new twist to these old traditions. AI as a cultural drive needs to be seen in the context of post-industrial life, in which we are, as described in Chapter 3, constantly surrounded by, interfaced with, and defined through machines. At its worst, AI adds a layer of seductive familiarity to that machinery, sucking us into a mythology of user-friendliness and humanity while the same drives of efficiency, predictability, quantifiability, and control lurk just beneath our perception.

But at its best, AI invokes a hope that is recognizable to humanists — that is invoked, in fact, by Donna Haraway in her "Cyborg Manifesto" [Haraway, 1990a]. This is the hope that, now that we are seemingly inescapably surrounded by technology, this technology can itself become hybridized and develop a human face.[4] This version of the AI dream is not about the mechanistic and optimized reproduction of living creatures, but about the becoming-living of machines. The hope is that rather than forcing humans to interface with machines, those machines may learn to interface with us, to present themselves in such a way that they do not drain us of our humanity, but instead themselves become humanized.

> *AI has a documented history of building military technology and mechanical replacements for human workers. Neither of these goals are ones that many cultural theorists would feel comfortable with. How does your project situate itself within this history?*

It is true that AI has a long and rich history of being used in ways with which cultural theorists generally might not agree. But, like many cultural practices, it cannot be summed up by its dominant uses; AI includes a heterogeneity of viewpoints and purposes. The technical application I work on here is in the subfield termed 'AI, art, and entertainment.' Application domains in this area run the gamut from automated sales representatives to interactive virtual pets to serious attempts at art; compared to the generation of robotic helicopters for the Department of Defense, these applications have, at least until now, been relatively innocuous.

I do believe, however, that AI research cannot proceed without awareness of how the techniques it develops are used in practice, whether or not one personally works on those applications that may be disagreeable. I also believe that this awareness is not particularly well-developed in my work, in any sense other than the relatively common AI strategy: I

[4]This is not to deny that one might want to resist mechanization — it simply bows to the reality that it will probably be a long time before such resistance could bear substantial fruit.

did my best to make my application be one I was willing to stand behind without qualms, and I tried (I think successfully) not to allow my own Department of Defense funding to alter the way in which I did and presented my work.

My own goal with respect to these practices was not to enable or disenable any particular application domain, but to try to develop a strategy for AI research where the application and funding of the research itself can be brought onto the table. Because agents are often seen as existing in a sociocultural vacuum, questions about funding and application are currently seen as ethical questions, to be sure, but ones that come after the fact and do not have a real implication for how research is conducted in the first place. I have tried to replace this model of research with one where the implications of the sociocultural context are made clear as part of the agent design, so that these 'external' questions can be seen for what they really are: at least partially constitutive of the way in which research can be done at all. This is admittedly a first step, but not, I think, a trivial one.

More broadly, I follow Jaron Lanier and J. MacGregor Wise in believing that one of the major dangers inherent in the way we build agents (and indeed, many technical artifacts) today is in the myth of authorlessness that surrounds their construction [Lanier, 1996] [Wise, 1996]. Agents are the creations of human beings, and therefore will always have limitations, some of which can be clearly understood, and some of which are implicit in nontransparent ways in the details of the construction of the technology. The danger of presenting these artifacts as living, independent beings rather than as human products is that the decisions which its human designer made become invisible and therefore unquestionable. The notion of agent-as-autonomous in this sense unintentionally closes off the possibility of critique.

My conviction with respect to this problem is that the users of technology should not be given a technical artifact as a *fait accompli*, but should be able to have a level of critical engagement with the technology. This means the technology and its context should be constructed so that they allow the users to understand how they are being led to interact with the computer and each other in specific ways. This is in fact the rationale behind the user interface design of the Industrial Graveyard: the cartooniness emphasizes that the system was built by a human, and the lack of buttons the user can press reflects the constraints I explicitly put on the user in terms of their interactions. In general, users should be able to realize intuitively on the basis of the software design that any particular technology provides not only possibilities but also constraints, constraints which are often grounded in the culturally-based assumptions of the people constructing the technology. In short, users should be able to understand, too, that technology is not just a set of pre-given tools, but itself social, cultural, and changeable.

The Cultural Studies / AI Hybrid

Now we have come to the end. Before we part ways, I must cash in the promises I made in the introduction when I asked you to consider the most important purpose of this thesis, the synthesis of cultural studies with AI.

From an AI perspective, I said that the use of cultural studies within AI could lead to new and perhaps better technology. In this thesis, that technology is the Expressivator, an architecture for supporting the user in interpretation of agent behavior by providing narrative cues. This technology is different from current technology because it is based on a different conception of what agents fundamentally are, a conception that stems from cultural studies analyses. Cultural analysis brings in concepts that helped to make the Expressivator possible and that would have been difficult to develop from within the field of AI alone.

From a cultural studies perspective, I described two advantages of using cultural studies in a practice of AI. The first is that by actually practicing AI, the cultural critic has access to a kind of experiential knowledge of science that is difficult to get otherwise, and will deepen his or her theoretical analysis. This increased knowledge is expressed in two ways in this thesis: (1) the analysis of behavior-based AI as a manifestation of industrial culture in Chapter 3, and (2) the analysis of the metaphorical basis of behavior-based AI even into the details of the technology, which occurs throughout the thesis.

The second advantage is that working within AI allows cultural theorists to not only criticize its workings, but to actually see changes made in practice on the basis of those criticisms. The Expressivator reflects the cultural studies analysis in the fundamental changes it makes in how an agent is conceived and structured. This brings home at a technical level the idea that agents are not simply beings that exist independently, but have authors and audiences by which and for which they are constructed.

Finally, the common advantage I peddled for my approach is the potential alteration to the rhetoric of mutual assured destruction that currently seems to be prevalent in interdisciplinary exchanges between cultural studies and science. At the most direct level, the possibilities for communication are enhanced among readers who, whatever their background, now share a common set of concepts which include, on the one hand, AI terms such as behavior-based AI, autonomous agents, and action-selection, and, on the other, cultural theory concepts such as objective vs. subjective technology, schizophrenia, and atomization. But the most fundamental contribution this thesis tries to make toward a cease-fire in the Science Wars is in demonstrating that 'science criticism' is relevant to and can be embodied in the development of technology, so that there are grounds for the two sides to respect each other, as well as a reason for them to talk. My hope is that this thesis can join other similarly motivated work on whatever side of the interdisciplinary divide to replace the Science Wars with the Science Debates, a sometimes contentious and always invigorating medley of humanist, scientific, and hybrid voices.

Appendix A

Technical Details

This appendix gives further details about how the Expressivator is implemented. Section A.1 describes the implementation of signs, signifiers, and the sign-management system. Section A.2 describes the changes to the Hap language that are needed to invoke meta-level controls, and how each language change was implemented. Section A.3 gives the details on the implementation of transition triggers and transition demons. Finally, section A.4 summarizes the changes made to the Hap language in Chapters 5 and 7.

A.1 Details of Sign(ifier) Implementation

As explained in Chapter 5 (pp. 113 - 121), sign management is a technique for structuring the agent in terms of the agent's impression, rather than in terms of internalistic problem-solving. There are three layers to agent structure under sign management — *signs*, which are small sets of physical actions that are likely to be interpreted in a particular way by the user; *low-level signifiers*, which are units of signs, physical actions, and mental actions (arbitrary C code) which communicate particular immediate physical activities to the user; and *high-level signifiers*, which communicate the agent's high-level activities.

Because the interpretation of agent activity depends heavily upon context, signs and signifiers are identified by the designer not when their code is defined, but in the context in which they are used. The same set of physical actions may be a sign in one context, and no sign or a different sign in another context. Similarly, a behavior may be a low-level or high-level signifier in one context, and no signifier or a different signifier in a different context. Signs are identified when they are posted (see below). Signifiers are identified by special annotations in the behavior language when the behavior is invoked: low-level signifiers with `low_level_signifying`, high-level signifiers with `high_level_signifying`. The 'mope-by-fence' signifier, for example, is invoked as `(with low_level_signifying (subgoal mope_by_fence))`.

These annotations mark the given behavior as a low- or high-level signifier, enabling their proper manipulation by other special forms. For example, since the same behavior can be a low-level signifier in some uses, and a regular behavior in others, the form that posts the low-level signifier checks to make sure the behavior *is* a low-level signifier in the current usage before it posts it. Marking behaviors as signifiers also enables the designer to write code that tests whether behaviors are low- or high-level signifiers (see section A.1.2 below). This property will become crucial in later code examples.

A.1.1 Posting Signs and Signifiers

In addition to allowing the designer to structure the agent according to these units, the sign-management system supports structures so that the *agent* can keep track of the signs and signifiers it has communicated to the audience. Signs and signifiers are stored in special data structures, described below. The agent *posts* its signs and signifiers when it is confident they have been communicated. It does this through special `post_sign`, `post_low_level_signifier` and `post_high_level_signifier` forms, which modify the sign and signifier data structures.

Sign / Signifier Data Structures

At any point in time, the agent will have at most one high-level signifier posted. Which high-level signifier is currently posted (i.e., has been demonstrated to the user) is noted in global memory in the working memory element called `CurrentHighLevelSignifier`, which has two fields: `name` (the name of the signifier) and `time` (the time when the signifier was posted).

The agent will usually have only one high-level signifier running.[1] But since high-level signifiers can only be posted once they have been communicated to the user, the currently *running* high-level signifier is often not the same as the currently *posted* high-level signifier. A high-level signifier may be active for quite some time before it is posted as the CurrentHighLevelSignifier (or may, if interrupted, never be posted at all).

Since signifiers are behaviors, both low-level and high-level signifiers are stored just like any other Hap behavior, in special working memory elements called 'Goal' with pointers to their parents and children. The sign and signifier data structure is an addition to this already-existing structure, in order to allow related high-level signifiers, low-level signifiers, and signs ready access to one another.

Signs and signifiers are stored in memory as shown in Figure A.1. A high-level signifier stores the name of its currently-posted low-level signifier and a pointer to its currently-posted sign.[2] A high-level signifier may have more than one active low-level signifier as a child (for example, during transitions), but each currently active high-level signifier only stores the name of *one* of those low-level signifiers, i.e. the one that has been most recently posted.

Low-level signifiers, in turn, store pointers to the high-level signifier of which they are a part — whether or not either signifier has been posted. This makes it easier to implement the posting of low-level signifiers, since

[1] It sometimes has more than one, for example during transitions.

[2] Logically, it would have made more sense to store the sign on the low-level signifier. I did not do this because I implemented signs before I realized I needed low-level signifiers.

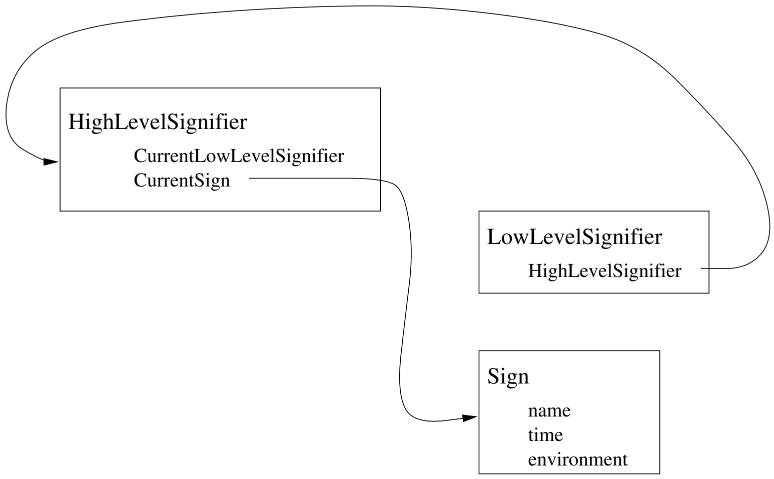


FIGURE A.1: Sign and Signifier Data Structures

```
(sequential_production read_lines (bottom current
                                   increment)
  (subgoal read_line $$current)
  (post_sign read_line ((line_read $$current))
  (subgoal continue_read_lines $$bottom
           "$$current + $$increment"
           $$increment))
```

FIGURE A.2: Example of `post_sign`.

they can easily find the high-level signifier to which they belong, even when (as is regularly the case) that high-level signifier is not posted to global memory yet. Signs simply store their own information: their name, the time they were posted, and a field, `environment`, that stores their arguments as a first-class environment.

Special Forms for Posting Signs and Signifiers

There are three forms for posting signs and signifiers: `post_sign`, `post_low_level_signifier`, and `post_high_level_signifier`. They are responsible for updating the data structures described in the previous section so that they remain a consistent picture of what the user has seen the agent do.

`post_sign`

The `post_sign` form takes as argument an arbitrary label and an optional first-class environment that contains the arguments of the sign. For example, Figure A.2 shows how the Patient reads the lines of the schedule. After each line is read, the Patient posts a sign that reminds itself which line the user has seen the Patient read.

When invoked, `post_sign` creates the 'Sign' data structure, a working memory element which includes the sign's name, its arguments, and

```
(parallel_production goto_spot (x y)
  (subgoal face_then_goto $$x $$y)
  (with persistent
    (demon
       ;;
       ;; check if 'goto_spot' is a low-level
       ;; signifier
       ;;
       (("G (Goal name == goto_spot;
                  low_level_signifying_p
                                  == true;);"
         ;;
         ;; check that 'goto_spot' is not posted
         ;; as a low-level signifier
         ;;
         "- (Goal CurrentLowLevelSignifier
                     == goto_spot;);"
         ;;
         ;; check that the 'walking_to' sign has
         ;; been posted
         ;;
         "CS (CurrentSign
                  name == walking_to;);"))
      (post_low_level_signifier goto_spot))))
```

FIGURE A.3: Example of `post_low_level_signifier`

a time stamp. It then notes the sign on the high-level signifier which invoked the `post_sign` form (this may be the CurrentHighLevelSignifier, but may also be a different, as-yet-unposted signifier). In this case, the 'read_lines' behavior is part of the 'read-schedule' high-level signifier, so it will make 'read_line' the current sign for 'read-schedule,' replacing whatever sign had previously been stored.

`post_low_level_signifier`

The `post_low_level_signifier` form works similarly, but it only takes the low-level signifier's name (no arguments). Its responsibility is to update the current high-level signifier's data structure so that its `CurrentLowLevelSignifier` field has the name of this low-level signifier. For example, the code fragment that implements going to a particular spot in Figure A.3 uses `post_low_level_signifier` to post the 'goto_spot' low-level signifying behavior on its parent high-level signifier (which happens to be 'explore_world') after the 'walking_to' sign has been posted.

An important note: it is not enough for a signifier to post itself once, when it is first demonstrated to the user. This is because behaviors can be interrupted; and a signifier that is interrupted may no longer be posted when control returns to it. It will need to post itself again after it is, once again, demonstrated to the user. Signifiers therefore continuously repost themselves whenever they see the appropriate signs or signifiers

```
(parallel_production mope_by_fence ()
  (with persistent (priority_modifier 100)
    (subgoal sad_looks_through_fence_to_sigh_demon
             $$this_plan))
  (with low_level_signifying
    (subgoal sad_looks_through_fence))
  (with persistent
     (demon
          ;;
          ;; check that sad_looks_through_fence is
          ;; the current low-level signifier
          ;;
        (("G (Goal CurrentLowLevelSignifier
                    == sad_looks_through_fence;);"
          ;;
          ;; check that mope_by_fence is not posted
          ;;
          "- (CurrentHighLevelSignifier
                   name == mope_by_fence;);"))
        (post_high_level_signifier mope_by_fence)))
  (wait))
```

FIGURE A.4: Example of `post_high_level_signifier`

and notice they are not currently posted.

`post_high_level_signifier`

`post_high_level_signifier` works in an way that is analogous to `post_low_level_signifier`. It modifies the working memory element `CurrentHighLevelSignifier` to hold the name of this high-level signifier. For example, the code fragment for 'mope_by_fence' in Figure A.4 waits until the 'sad_looks_through_fence' low-level signifier has been posted, and then posts the high-level signifier 'mope_by_fence.'

Final Note

I set up the system so that information about signs and low-level signifiers are stored on the high-level signifier of which they are a part. After implementing the Patient, it became clear that they should be posted on global memory instead, since you sometimes want to know what the last sign or low-level signifier was even after the high-level signifier that posted them is gone. Certainly, it is possible to have multiple low-level signifiers and signs be posted simultaneously to different high-level signifiers, but in practice this property was not particularly relevant to the running of behavior. It seemed that merely overwriting the most-recently-posted signifier or sign — no matter which signifier it had originated from — would have been simpler to implement and just as useful.

```
(sequential_production show_reaction_to_line ()
  (locals (current_line 0))
  ;;
  ;; find out what line the user saw me read
  ;;
  (with (success_test
            ("CS (CurrentSign name == read_line;
                              $$current :: c;);")
                  ;;
                  ;; store it in a local variable
                  ;;
            (:code "$$current_line = c; "))
      (wait))
  ;;
  ;; show a reaction to that line
  ;;
  (subgoal show_reaction $$current_line))
```

FIGURE A.5: Sign variables can be matched as part of the `CurrentSign` wme by preceding the name of the variable with `$$`
.

A.1.2 Matching on Signs and Signifiers

As you may have noticed from the previous code examples, behaviors can match on signifiers just as on anything else in memory. The name of the high-level signifier can be found on the `CurrentHighLevelSignifier` WME, and the name of the high-level signifier's low-level signifier can be found as a field of that high-level signifier. Since signifiers are behaviors, they can also be matched as any other behavior can; they can be distinguished from other behaviors using the flags `low_level_signifying_p` and `high_level_signifying_p`.

Signs can be found on the `CurrentSign` WME as described above. A special property of signs is that they include not only a name but a first-class environment which represents their arguments. These arguments can be matched straightforwardly through a special syntax which is shown in FigureA.5. This code fragment is taken from a transition; it checks which line of the schedule the user has seen the Patient read, then shows a reaction to that line. The `$$` syntax is unpacked by the Expressivator compiler and used to generate matching code for the proper variable in the `CurrentSign`'s first-class environment.

A.2 Details of Meta-Level Control Implementation

Meta-level controls are introduced in Chapter 5 (pp. 124 - 127). They are special powers that behaviors can use to find out about and coordinate with each other. Meta-level controls are implementable in many behavior-based architectures. This section describes how they are implemented on top of Hap for the Expressivator. They involve the following changes to the way Hap works:

1. *Querying behaviors*: I make use of the as-yet-underutilized Hap behavior matching as a regular part of the Expressivator. I add `low_level_signifying_p` and `high_level_signifying_p` as fields to behaviors, so that other behaviors can test for them.

2. *Deleting behaviors*: I add the primitives `succeed_behavior` and `fail_behavior` to allow behaviors to terminate other behaviors either successfully or unsuccessfully.

3. *Invoking higher-level behaviors*: I generalize Loyall's `breed_goal` function to allow behaviors to add new subbehaviors to any other behavior, or to the agent's top level.

4. *Adding new subbehaviors to other behaviors*: Loyall's `breed_goal`, as adapted for invoking higher-level behaviors, is also used to add new subbehaviors to other behaviors.

5. *Changing internal variables*: I add the concept of Communicative Features and a data structure to store them. Communicative Features allow behaviors to coordinate their presentation in order to present a coherent picture to the user.

6. *Paralyzing behaviors*: I add the primitives `turn_on_muscles` and `turn_off_muscles` to allow behaviors to paralyze and unparalyze other behaviors.

7. *Moving running subbehaviors*: I add `succeed_and_strip_behavior` and `fail_and_strip_behavior` primitives to allow subbehaviors to be switched to another behavior, while causing the old behavior to believe the now-missing subbehavior either succeeded or failed, respectively.

Each of these changes is discussed in more detail in the sections that follow.

A.2.1 Querying Behaviors

In Hap, behaviors sense the environment and check structures in memory by matching against RAL working memory elements (for more information on RAL, see [Forgy, 1991]). For example, the RAL test

```
"B (BelievedLocation x == $$x1;
                     y == $$y1;);"
```

checks the x and y values of the `BelievedLocation` WME against the value of the variables `$$x1` and `$$y1`. The RAL test

```
"P (PositionSensor who == I;
                   valid == true;
                   who_x == $$x;
                   who_y == $$y;
                   xydist != -1;
                   xydist <= RADIUS_ERROR);"
```

senses the position of the agent and checks whether it is within a small distance of the target point `($$x,$$y)`.

In order to be able to sense other behaviors, then, behaviors need to be represented as RAL WMEs against which behaviors can match, in the same way they do for data in memory or environmental conditions. This turned out to serendipitously already be the case in Hap, as a side-effect of its implementation. The compiler turns behaviors into WMEs called 'Goal,'[3] which include the field `name`, which is used most often in testing.

This attribute of Hap was not actually used for anything (or, for that matter, common knowledge among Hap users) until Bryan Loyall used it as a basis for adding a meta-level control, `breed_goal` (which will be discussed below) in the version of Hap he implemented for his thesis [Loyall, 1997a]. In order to be able to sense behaviors in the Expressivator, the only change that was necessary was to add the fields `low_level_signifying_p` and `high_level_signifying_p` to the 'Goal' WME so that behaviors can sense whether behaviors are signifiers. The `CurrentSign` and `CurrentLowLevelSignifier` fields mentioned above are also implemented as part of the Goal WME.

Once behaviors are matched, they often need to be stored and passed around. For example, a behavior may try to find out which low-level signifier is currently running, then tell one of its subgoals to delete that low-level signifier. The 'Goal' WME, which represents a behavior, stores an integer pointer to itself in the field `self`. This integer is used to refer to behaviors by the meta-level controls that follow (for an example, see Figure A.6).

A.2.2 Deleting Behaviors

The underlying Hap architecture has always needed to terminate behaviors; the change in the Expressivator is to make this internal function also available for behaviors to call directly as part of the behavior language. Specifically, I added `succeed_behavior` and `fail_behavior` primitives that took as arguments a behavior pointer, and would terminate that behavior either successfully or unsuccessfully (see Figure A.6).

It turned out in practice that being able to terminate a particular, specified behavior was not that useful for transitions. This is because transitions take place from one externally-seen signifier to another. This externally-seen signifying behavior may not be the same as the internal behavior the agent is engaging in, for example if the agent just changed to a new signifier but has not emitted its signs yet. In this case, the transition does not want to kill the externally-seen but no-longer-existent signifier from which it ostensibly comes. Rather, it will want to kill the newly-begun-but-not-yet-announced signifier which the transition will replace.

The solution to this problem is to introduce new forms which terminate, not a particular signifier, but *any* signifier which is in conflict with this one. Specifically, `kill_low_level_signifier`, when called within a particular low-level signifier, terminates (successfully) all other

[3] Hap makes a distinction between 'goals' (the name of a behavior) and 'behaviors' or 'plans' (the way in which behaviors are implemented), which is not pertinent to the current discussion. I have left it out for fear that it will hopelessly muddy the discussion.

```
(sequential_production
      freeze_in_place_interrupt (parent_plan)
  (locals (g 0))
  ;;
  ;; whirl around
  ;;
  (subgoal whirl_around)
  ;;
  ;; add low-level signifier 'freeze_in_place'
  ;; to the high-level signifier
  ;;
  (breed_goal $$parent_plan freeze_in_place)
  (subgoal wait_for "random_range(1000,6000)")
  ;;
  ;; finish freezing after you have waited a
  ;; while
  ;;
  (demon (("G (Goal name == freeze_in_place;
                  self :: s;);")
          (:code "$$g = s;"))
    (succeed_behavior $$g)))
```

FIGURE A.6: Example of use of matching on behaviors, `breed_goal`, and `succeed_behavior`

low-level signifiers — whether posted or not — that are part of the same high-level signifier as the calling behavior (i.e., all of that behavior's low-level siblings). `kill_high_level_signifier` similarly terminates (successfully) all other high-level signifiers. Both of these forms are implemented as behaviors.

A.2.3 Invoking Higher-Level Behaviors

Invoking new behaviors is a normal function of Hap. Any behavior can generate new subbehaviors. But transition behaviors need to add, not subbehaviors, but new behaviors at *higher* levels.

Specifically, if a transition behavior starts up a new low-level signifier as a subbehavior, the low-level signifier will be the 'child' of the transition behavior rather than of its 'real' high-level signifier parent. Because of the semantics of Hap, this also means the transition behavior needs to stick around until the signifier it invokes terminates, which seems wrong; the transition behavior should be done as soon as the new signifier begins, not when it ends.

These concerns mean that the transition should invoke the new subbehavior, not as part of itself, but as part of its parent high-level signifier (if it is a low-level signifier) or part of the agent's top-level behavior (if it is a high-level signifier). Fortunately for me, Loyall implemented a `breed_goal` form that does this as part of his thesis work. It is limited, however, in that it only works for adding subbehaviors to parallel behaviors, i.e. behaviors whose subbehaviors all run simultaneously. For the Expressivator, `breed_goal` is generalized to work for all behaviors,

```
(parallel_production
     read_sign_to_exercise_transition_demon ()
  (locals (exercisep 0))
  ....
  (breed_goal $$apt_plan sb_exercise)
  (demon (("GE (Goal name == exercise;
                  has_child == true;
                  self :: s;);")
          (:code "$$exercisep = ...."
                 ;; plan for exercise behavior
          ))
    (breed_goal $$exercisep watch $$overseer)))
```

FIGURE A.7: Example of using `breed_goal` to add a new subbehavior to another behavior.

whether parallel or sequential; the new subbehavior will run in parallel with the behavior's original subbehaviors. You can see an instance of `breed_goal` in practice in Figure A.6.[4]

A.2.4 Adding New Subbehaviors

Loyall's `breed_goal` can be used in the straightforward way for adding new subbehaviors to a specified behavior. Figure A.7 shows how the transition from reading the schedule to exercising uses `breed_goal` to add to the new exercise behavior a subbehavior to watch the Overseer.

A.2.5 Changing Internal Variables

Neal Reilly [Neal Reilly, 1996] added Behavioral Features to Hap in order to ease the problem of behavioral coordination. Behavioral Features are variables like "aggression" or "fear" that behaviors share. They are somewhat like emotions, but rather than representing how the agent 'feels,' they represent how behaviors should display the agent's emotions. For example, one agent, when afraid, may become aggressive; another may become quiet and shy.

I used the same basic mechanism as Behavioral Features, but I termed them Communicative Features to make clear that they are things that the behaviors need to communicate to the audience. Communicative Features are stored in a special data structure on global memory which includes two fields: an arbitrary label, `type`, and an integer `intensity`.

In the Industrial Graveyard, I used two Communicative Features: fear and woe. Although this was not my original intention, they correspond basically to a kind of simple emotional system. Whenever a traumatic event happens, the transition into the event calls a function "traumatize" that increases the agent's fear. After a traumatic event, as the fear subsides, woe increases. If the agent is left alone for a long time, woe goes back down. Sadly for the little agent, woe is usually maxed out by the

[4] Technical detail that is meaningless to all but Hap cognoscenti: `breed_goal` takes as an argument, not an integer behavior pointer, but an integer plan pointer.

```
;;
;; This behavior causes the Patient to tremble
;; when the Overseer is near it.
;;
(sequential_production
                 tremble_overseer_when_close ()
  (locals (feardist "0"))
  ;;
  ;; find out how scared I am supposed to look
  ;;
  (demon (("CF (CommunicativeFeature
                   type == fear_of_overseer;
                   intensity :: i;);")
          ;;
          ;; make distance at which to tremble
          ;; short if not scared, long if scared
          ;;
          (:code "$$feardist = 350 * (i / 5);"))
    ;;
    ;; tremble when Overseer is less than this
    ;; distance away from me
    ;;
    (subgoal tremble_overseer_at_dist
             $$feardist))))
```

FIGURE A.8: Example of use of Communicative Features

end of the story. I also used distance from Overseer to influence fear, but it did not affect the woe.

Despite the similarity with Neal Reilly's system, for me the function of these 'emotions' is not so much *as* emotions — although they do influence the agent's behavioral choices — but as a way to knit together disparate behaviors. That is, the Communicative Features act as a kind of behavioral smoothing between behaviors. Without the Communicative Features, the agent might go from a totally miserable round of moping to a very cheerful hop across the room, which looks very wrong. With my two features influencing most of the behaviors (this took about two days to add — for an example see Figure A.8), the behavioral consistency looked much better.

Having gotten this 'emotion system' to work by making it maximally simple, I suspect that a complex emotional system is not appropriate for really expressive agents. This may sound like a paradox. But just as behaviors can be hard to understand if you cannot see what is motivating them, subtle emotions are difficult to understand unless you clearly explain what is making the emotions arise. For example, originally I just had the agent's fear go up when the Overseer came close, and the fear go down when the Overseer went away. I found this made the system hard to understand, because the agent's emotions would change without the agent necessarily displaying any reaction to the event causing the emotional damage. This is why I choose to traumatize the agent when

```
(sequential_production
          paralyze_high_level_signifier ()
  ;;
  ;; find a behavior which is a high-level
  ;; signifier
  ;;
  (precondition
      ("G2 (Goal high_level_signifying_p == true;
                 self :: s);"))
  (locals ("g" "s"))
  (turn_off_muscles $$g))
```

FIGURE A.9: Example of turn_off_muscles

it is reacting to the Overseer, rather than when it senses the Overseer — this links the emotional change clearly with what the user is seeing.

A.2.6 Paralyzing Behaviors

By 'paralyzing' a behavior, I mean allowing a behavior to run while intercepting all of its muscle commands. This means behaviors can have effects in Communicative Features, but not in actual movement. I implemented this by using dummy movement commands that check to see if a behavior has its muscles turned on before actually doing the movement. Any behavior can turn on or off the muscles of any other behavior using the constructs `turn_on_muscles` and `turn_off_muscles` (for an example, see Figure A.9).

A.2.7 Moving Running Subbehaviors

Conceptually speaking, moving subbehaviors while they are running is straightforward. The behavior is simply taken from its parent and reinstalled under a different behavior. The `succeed_and_strip_behavior` and `fail_and_strip_behavior` primitives do just this: they move a given subbehavior from one behavior to another, while causing the former parent behavior to believe the suddenly disappeared subbehavior has succeeded or failed, respectively.

While this is conceptually simple, it was technically the most complex meta-level control to add. It basically corresponds to doing brain surgery on the agents. Since the compiler never expected behaviors to move around while they were running, when subbehaviors are taken out from one place and moved to another there is a large and not clearly marked (not to mention largely uncommented) group of pointers that need to be reinitialized to their new, proper values.

In practice, I found that this meta-level control was not really worth the enormous effort it took. Moving subbehaviors was better dealt with by simply deleting the old subbehavior and starting a new version of the same subbehavior in the new spot. In fact, after all the work I put in it, I did not end up using this meta-level control at all!

A.2.8 Related Work

As mentioned on page 125, a number of meta-level controls already exist in other behavior-based architectures. Brooks introduces the idea of subsuming behavior's action commands; Neal Reilly introduces Behavioral Features; Blumberg has Internal Variables and meta-level commands. The meta-level control system here attempts to bring some order to these features by finding a small set that will support behavior transitions.

Meta-level controls are reminiscent of metalevel plans in PRS[Georgeff and Ingrand, 1989]. Like metalevel plans, meta-level controls are intended to allow behaviors to use and manipulate meta information about the system's processing. However, PRS's metalevel plans are intended to be used to allow the system to plan its otherwise reactive behavior, and concentrate on formalizing the system's self-knowledge. Meta-level controls are intended to help designers coordinate behaviors, and focus on adding just enough power so the designer can write behaviors that explicitly refer to one another.

A.3 Details of Transition Implementation

Once meta-level controls are implemented, most of what you need to implement transition triggers and transition demons is already available. In addition to the meta-level controls, I made the following changes to Hap for the Expressivator:

- I added a data structure so transition triggers and transition demons could share information about transitions.
- I added `create_mini_transition` and `create_maxi_transition` primitives to create the transition demons.

Here, I will describe the transition data structure and the implementation of transition triggers and transition demons.

A.3.1 Transition Data Structure

Transition data is stored in the `Transition` WME. Mini-transitions are created with and stored on the high-level signifier to which they belong; the one and only maxi-transition is stored on global memory. The following data is stored in the `Transition` WME:

- **to**: which signifier is being switched to
- **from**: which signifier is being switched from
- **reason**: an arbitrary label which is selected by the designer to represent the reason for the transition (and, hence, what the transition demon must demonstrate)
- **valid**: has value 1 iff the transition has been triggered, but has not been implemented by a transition demon yet
- **switching**: has value 1 iff the transition is in process. It is automatically turned off when the next signifier is posted.

- **type**: whether it is a mini- or maxi-transition
- **high-level signifier**: for mini-transitions, lists the name of the high-level signifier to which this mini-transition belongs.

A.3.2 The Gritty Details of Transition Trigger Implementation

At their most basic, the job of a transition trigger is to notice when it is time to change behaviors for a particular reason. A transition trigger generally runs in the background, waiting for the right combination of environmental factors, signs, signifiers, etc. When a transition trigger notices that it is time to change behaviors, it notifies the rest of the system by altering the `Transition` data structure. Transition demons will check those data structures and fire to implement the transition. An example of a transition trigger is shown in Figure A.10.

Triggers generally want to fire only when particular behaviors are or are not being engaged in. Sigh, for example, only wants to fire when the agent is feeling sad, not feeling very afraid, and is not engaging in react-overseer or a similarly urgent, hyper behavior. Typically, then, triggers are complex demons that go on the alert when an appropriate behavior to switch from is happening, and then have conditions that abort the alert when the behavior is no longer happening.

Triggers turn out to be complicated at times because signifiers become internally active before the user notices them (i.e., before they are posted). Sometimes, triggers need to fire off of what is going on internally, while at other times, what matters is what the user has seen. The actual conditions under which the trigger should fire must be thought out carefully.

For example, when headbanging, if the light goes on the Patient should kill the smack-head behavior immediately, even if it has not been posted yet. This is because smack-head will hit the Patient's head on the floor before it can post its first signifier, and the lamp looks pretty unreactive if it hits its head when the light is on. So even before smack-head has been posted, the transition trigger must be on the lookout for possibly transitioning out of it.

On the other hand, the transition sequence *itself* needs to move from user-seen behavior to user-seen behavior. If smack-head is active, but has not been posted yet, the user will still think the Patient is in its wait-for-light-on behavior. The transition that will be demonstrated must go from that externally seen wait-for-light-on behavior, not the internally active smack-head behavior!

I found it helpful in designing triggers to think in terms of durations under which different conditions are true:

```
;; This is the trigger for the transition from
;; headbanging to being killed.  It fires when
;; the Overseer comes near.
(sequential_production
        monitor_overseer_approach_to_be_killed ()
  (with (success_test
         ;; check that this transition has not
         ;; already fired
         ("TT (Transition type == maxi;);"
          "- (Transition
                 type == maxi;
                 from == head_banging;
                 reason == overseer_approached;
                 switching == 1;);"
          "S (Self me :: I;);"
          ;; check that it is time for the
          ;; patient to die
          "SS (StoryStage stage == SS_DIES;);"
          ;; check that the user knows I am
          ;; headbanging
          "CHS (CurrentHighLevelSignifier
                        name == head_banging;);"
          ;; check that the Overseer is near me
          "PS (PositionSensor
                     who == I;
                     valid == true;
                     target_who == $$overseer;
                     xydist > -1;
                     xydist < 150;);")
          "ESPosition, make_position_wme,
              modify_position_wme, 6, self, -1,
              -1, -1, $$overseer, -1"
          ;;
          ;; Trigger the transition to be_killed
          ;;
          (:code
            "modify TT t {
               t->to = be_killed;
               t->from = head_banging;
               t->reason = overseer_approached;
               t->valid = 1;};"))
        (wait)))
```

FIGURE A.10: An Example of a Transition Trigger

```
;;
;; This is a transition from reacting to the
;; overseer to stepping around the environment
(parallel_production react_to_step_demon
                     (parent_plan)
  ;;
  ;; transition trigger: wait for me to be
  ;; reacting to the Overseer, and for the
  ;; Overseer to go away
  ;;
  (with (persistent when_fails)
    (subgoal check_when_overseer_goes_demon))
  ;;
  ;; transition demon:
  ;;
  (create_mini_transition
          (step parent_plan
                "reason == overseer_goes;"
                :from react_overseer)
    ;;
    ;; kill whatever signifier came before me
    ;;
    (subgoal kill_low_level_signifier)
    ;;
    ;; stop cowering
    ;;
    (act "AStopTremble")
    (act "AStopLook")
    (act "AStopFace")
    (subgoal stop_crouching)))
```

FIGURE A.11: An example of `create_mini_transition`

<table>
<tr><td>Old signifier running</td><td>Transition triggers</td><td>Transition demon starts</td><td>New signifier starts</td><td>New signifier posts</td></tr>
<tr><td colspan="4">Old signifier posted</td><td>New signifier posted</td></tr>
<tr><td></td><td>Transition valid</td><td colspan="3"></td></tr>
<tr><td colspan="2"></td><td colspan="2">Transition switching</td><td></td></tr>
<tr><td colspan="2">Old signifier runs</td><td>Transition runs</td><td colspan="2">New signifier runs</td></tr>
</table>

Depending on the situation, it would be appropriate to trigger off of almost any of these changes in state. This definitely adds a level of complexity to designing the triggers properly.

A.3.3 The Similarly Horrendous Details of Transition Demon Implementation

In order to implement mini-transitions, I added a new form to Hap called `create_mini_transition`. This form is used to automatically set up most of the bookkeeping details that are involved with transition demons. The create-mini-transition form takes the following arguments:

- a variable representing the high-level signifier of the mini-transition (to which the new subbehavior should be attached),
- the name of the behavior to which the mini-transition switches,
- a piece of RAL code which tests the reason for the transition,
- the set of steps that make up the transition sequence.
- a set of optional, keyworded arguments, including
  - `:old_beh` for the behavior the transition is from
  - `:interrupt` if the transition is an interruption

(see Figure A.11 for an example).

The mini-transition then sets up a demon which checks for the transition to fire for the correct behavior and reason. This demon then calls another behavior which implements the transition. The transition can be implemented in one of two ways: (1) actually do the given transition sequence or (2) just kill the old behavior and jump directly to the new one (the 'sudden break' which is the norm in other agent architectures). The system does the first option most of the time, but will use the second option when transitions are turned off, or when the user is not actually looking at the agent.

The same basic technique is used for maxi-transitions (see Figure A.12).

A.4 Summary of Expressivator as Agent Language

Implementation of the Expressivator is spread through Chapters 5 and 7. Here, I summarize the changes made in both chapters to Hap as an agent programming language.

- Signs, Signifiers, and Sign Management
  - The markers `low_level_signifying` and `high_level_signifying` are added to the language in order to allow the declaration of low-level and high-level signifiers.
  - The form `post_sign`, along with an arbitrary list of variables and their values, allows signs to be posted in common memory with a timestamp and their variable list.
  - The forms `post_low_level_signifier` and `post_high_level_signifier` are added to the language. When invoked, they store the name of their enclosing low-level (respectively, high-level) signifier.

```
;; this is the transition from headbanging
;; to be-killed
(parallel_production
  headbanging_to_be_killed_transition_demon ()
  ;; transition trigger: fire when the Overseer
  ;; is coming to kill me
  (with persistent effect_only
    (subgoal
        monitor_overseer_approach_to_be_killed))
  ;; transition demon:
  (create_maxi_transition
     (be_killed "reason == overseer_approached;"
                :old_beh head_banging)
    ;;
    ;; kill whatever high level signifier is
    ;; running
    (subgoal kill_high_level_signifier)
    ;;
    ;; make sure my eyes are shut
    (par
     (subgoal close_eyes)
     ;;
     ;; stop - do you hear someone coming?
     (subgoal wait_for 500))
    (par
     ;;
     ;; traumatize myself
     (with effect_only (priority_modifier -5)
   (subgoal traumatize 5))
     (seq
      ;;
      ;; whirl around blindly
      (subgoal whirl_around)
      (subgoal wait_for 800)
      (subgoal whirl_around)
      (subgoal wait_for 800)
      ;;
      ;; switch to new behavior
      (breed_goal $$apt_plan be_killed)))))
```

FIGURE A.12: Example of create_maxi_transition

- Signs can be tested by checking the `CurrentSign` wme, which is attached to the high-level signifier of which the sign is a part. The compiler is changed to allow the values of the sign's variables to be tested in the same way as any other memory element.
- Low-level signifiers can be tested by checking the wme named `CurrentLowLevelSignifier`, which is attached to the high-level signifier of which the low-level signifier is a part.
- High-level signifiers can similarly be tested by checking the `CurrentHighLevelSignifier` wme, which is a global variable.

* Meta-Level Controls
  - The ability to sense behaviors is already a part of Hap; the Expressivator includes the addition of a number of fields to the behavioral data structure: `low_level_signifying_p`, `high_level_signifying_p`, `CurrentLowLevelSignifier`, `CurrentSign`. These allow various additional aspects of the behaviors to be tested.
  - Behaviors can delete other behaviors, causing them to either fail or succeed, by calling either `fail_behavior` or `succeed_behavior`, respectively.
  - Behaviors can add subbehaviors to other behaviors or at the agent's top level by calling `breed_goal`. This functionality is already present in Hap and allows subbehaviors to be added to behaviors whose subbehaviors run in parallel. It is expanded in the Expressivator to be applicable to behaviors whose subbehaviors run in parallel or sequentially (the new subbehavior will always run in parallel).
  - Behaviors can move around running subbehaviors, switching them from an old behavior to a new one, by calling `succeed_and_strip_behavior` or `fail_and_strip_behavior`. The first construct causes the old behavior to believe the subbehavior succeeded; the second construct causes the old behavior to believe the subbehavior failed. Since the old behavior is usually deleted right away, it generally does not matter which one of these are chosen.
  - Behaviors can paralyze and unparalyze other subbehaviors by using the `turn_on_muscles` and `turn_off_muscles` constructs.

* Transitions
  - `create_maxi_transition` is called with the name of the new high-level signifier, a token representing the reason for the transition, the behavioral steps that express the reasons for the transition, and a set of optional keywords including the name of the old behavior and a keyword designating the transition as an interruption. It generates the code to

trigger the demon's steps when the given reason is cited for behavioral change to the given new high-level signifier.

– `create_mini_transition` works on exactly the same principle, but for low-level signifiers.

Appendix B

Detailed Analysis of *Luxo, Jr.*

This appendix contains the details of the analysis of *Luxo, Jr.* in terms of the behaviors and transitions that can be seen in Luxo's actions. Note that this division into behaviors and transitions is not written in stone; it is just one reasonably good match.

Senior		Junior		Ball
Transitions	Behaviors	Transitions	Behaviors	
	stands still			
starts slowly, slowly puts more movement in	when ball stops, turns to look at it			comes in and bounces off of senior
	examines ball			
no transition, except slight stop	smacks ball off screen			

Senior		Junior		Ball
Transitions	Behaviors	Transitions	Behaviors	
here, “watching” is a kind of transition	watches			comes back
	stops smacks ball again			
				comes back again, rolls past Senior
	Senior follows ball, smoothly turns back to Junior (offscreen)			
Shock reaction and scoots back while looking off-screen				
			hops onto stage	
		wiggles butt, looks at ball		

Senior		Junior		Ball
Transitions	Behaviors	Transitions	Behaviors	
	looks at ball		looks at S	
	looks at J	alternation	looks at ball	
	watches J		wiggles butt and hops off	
	watches ball			comes back
			comes back	
		looks at ball		
		gets in position		
			smacks ball	
				hits cord
			“struck” reaction	
	hits ball away		looks at ball	
		stops		
		looks at ball		
		hunkers down		
	looks		jumps on ball	
	surprised			
			rides ball	
				pops

Senior		Junior		Ball
Transitions	Behaviors	Transitions	Behaviors	
	leans in more	looks around as though wondering what's going on decides what to do (looking)		
	looks at ball		rolls back	
		looks at ball		
			flips flat ball over	
		sits back and looks		
	shakes head		looks at S	
			looks deflated sighs	
alternation as a kind of transition	looks at ball	sighing hop		
	looks offscreen		hops off screen	
shock reaction	hops back	off screen (transition from S's behavior!)		

Senior		Junior		Ball
Transitions	Behaviors	Transitions	Behaviors	
	watches (supervise)			big ball comes on stage
			hops after (double hop in middle)	
looks at screen	shakes head			

Appendix C

Case Study: Full Design of the Patient

In this appendix, I step through the entire design process for the Patient character of the Industrial Graveyard.

C.1 Selecting High-Level Signifiers

As described in Chapter 7, the first step in the agent design process is to decide on the agent's high-level signifiers. The Patient was designed in the context of the Industrial Graveyard; its behaviors needed to support the plot of the story (as described in Intermezzo I on pp. 87-88), as well as enhance the user's understanding of the point of the system.

The Patient's high-level signifiers roughly parallel the story plot.

- *In Monitor*: Initially, the user needs to understand that the Patient is being processed mechanically. This is represented in the Industrial Graveyard by having the Patient be examined by the Overseer in a machine called the Monitor. The Monitor reduces the Patient's subjectivity to simple numerics: the user is notified that the Patient has an identification number and a numerically identified 'disease' (short circuit), and that its demerits are being tracked by the system. The Patient's first high-level signifier represents its behavior as it is being processed into the system.

- *Explore World*: Once the Overseer is done processing the Patient, it leaves and the Patient can begin exploring the 'world' (i.e. the junkyard). While exploring the world, the Patient is constantly sanctioned by the Overseer whenever its movements exceed proper bounds. This behavior in connection with the Overseer's reactions to it demonstrates to the user the Patient's helpless position in the world.

- *Read Sign*: There is a schedule of daily activities displayed in the junkyard. As the Patient wanders around, it notices the schedule and goes up to read it. The schedule again is intended to make clear to the user that the Patient's activities are structured for it, and that it has no choice but to do what is on the schedule.

- *Exercise*: Once 10:00 strikes, the Patient must exercise. This consists simply of rapidly bobbing up and down.

- *Mope By Fence*: After some time of being bullied by the Overseer, the Patient becomes depressed. The Patient engages in the Mope by Fence behavior by slowly walking over to the fence of the junkyard, and sadly looking out at the outside world, now forever beyond its grasp. This should be a behavior chock full of pathos.

- *Head-Banging*: The Patient has a short-circuit. This means its light goes out from time to time, leaving it blind. In order to remedy the situation, the Patient may shake its head; if that fails, the Patient will start smacking its head on the ground in order to fix the short. This behavior is designed to be as negative as possible in the eyes of the Overseer; it involves the most jerky body movement.

- *Turned Off*: Whenever the Patient has been misbehaving, the Overseer will come over and turn it off. The Turned-Off behavior consists of the Patient collapsing onto the ground into an unnatural position. After a few seconds, the Patient gets up again and continues on its way as the 'sedatives' wear off.

- *Be Killed*: After the Patient has gotten in enough trouble, the Overseer decides that it is more efficient to turn the Patient off than to continue to monitor it. While the Patient is being killed, it needs to act very frightened so that the user knows something unusual is happening.

- *Unknown Behavior*: This behavior is designed to test one of the transition types, the unknown transition (p. 124 of Chapter 5). The Unknown Behavior consists of simple and relatively meaningless background activity that the lamp can engage in when it is not sure what it should be doing.

C.2 High-level Signifier Decomposition

Once the high-level signifiers are identified, they need to be decomposed. High-level signifiers are broken up into a set of low-level signifiers, which represent the major activities that make up the high-level signifier. These low-level signifiers will later be connected with mini-transitions.

- In Monitor

  - *Be Mechanical*: In the beginning, the Patient reinforces the mechanistic propaganda the user has just read by acting completely mechanical. The Patient doesn't blink; it moves slowly and mechanically, and it does not visually track objects or the Overseer.

  - *Tremble and Watch Overseer*: Once the Patient notices the Overseer, it 'comes to life.' It tracks the Overseer's movements and trembles nervously.

  - *Look Around Scared*: When the Patient is not watching the Overseer, it starts to examine its environment. It is, however, still frightened, so it still trembles now and then and uses quick, jerky looks.
  - *Look Around Curiously*: Once the Patient has gotten used to the Monitor, it becomes curious. It gets closer to the front of the machine, and looks out into the junkyard. Its looks are slower and longer, and its gaze follows things in the environment.

- Explore World
  - *Looking Around*: This is the behavior the Patient uses when it is trying to decide where in the world it should go. It looks around for interesting spots. It should not pick such spots near the Overseer.
  - *Go To Spot*: The Patient walks determinedly, if fearfully, to the spot it has chosen. It looks mostly at spot but checks out the rest of the environment, too.
  - *Look Around*: Once it has gotten to a particular spot, it looks it over. This behavior has a focus of interest.
  - *Sigh*: Overcome with sadness, the Patient occasionally interrupts other behaviors with a sigh.
  - *React to Overseer*: Whenever the Overseer comes close, the Patient reacts to it by trembling and acting fearful.
  - *Freeze in Place*: Occasionally, the Patient's paranoia gets the better of it, and it interrupts its behavior to freeze in place and look around for danger.

- Mope By Fence
  - *Look Out At World*: The Patient sadly stands at the fence and slowly moves its gaze around the outside world.
  - *Sigh*: Ah, what pathos! Let your sadness escape, little creature!
  - *Walk Up and Down Fence*: Sometimes the Patient will move up and down the fence a little to find a better viewing position.

- Read Sign
  - *Read lines*: The Patient moves its head from left to right in a reading motion.
  - *React to lines*: Sigh, shake its head, or read a line more than once.

- Exercise
  - *Bob up and down*: Exercising consists simply of this bobbing up and down motion.

- Head-Banging
  - *Hit head on ground*: The Patient flings its head back and then whacks it into the floor of the junkyard.

  - *Wait to see if light went out*: The Patient pauses in its head-banging to see if its light has come on yet.
- Unknown Behavior
  - *Sigh*: As always, sighing is an essential part of the Patient's existence.
  - *Look Around*: Look around aimlessly, seeing what is going on around the Patient.
  - *Watch Overseer*: Keep an eye on the Patient's evil enemy.
- Be Killed
  - *Fear City*: The Patient needs to show that it is extremely frightened. This is like the trembling at the Overseer mentioned earlier, but even more extreme.
  - *Die*: When the Patient dies, it turns into a cardboard cut-out.
- Turned Off
  - *Be Turned Off*: Collapse and stay turned off for a while.

Each of these low-level signifiers was implemented separately. Note, however, that some of the high-level signifiers share the same low-level signifiers; in this case, the code for them was shared as well.

Composing Low-Level Signifiers with Mini-Transitions

Once I selected the signifiers for the Patient, it was time to connect them to form the Patient's complete behavior. The first step was to synthesize the low-level signifiers with mini-transitions in order to generate the high-level signifiers. In order to do this, for each high-level signifier I made a list of all possible mini-transitions between its low-level signifiers.

Fortunately, many of the possible transitions turned out to be impossible. For example, Be Mechanical is always the first behavior, and always leads to noticing the Overseer and becoming frightened. Therefore, there is no need to implement transitions from Be Mechanical to any other behavior.

Once the mini-transition list was whittled down, I enumerated reasons for each behavior change. For each reason, I also listed how that reason could be concretely be communicated to the user. These two aspects form the basis for the design and implementation of each mini-transition[1].

The synthesis of each high-level signifier through mini-transitions is laid out in Figures C.1 through C.10.

[1] In the case of the Patient, I contented myself to have only one reason for each behavior change, but there is no reason to limit oneself this way in general.

For **In Monitor**:
Low-level signifiers:

1. Be Mechanical
2. Tremble and Watch Overseer
3. Look Around Scared
4. Look Around Curious

Transitions:

From	**To**	**Reason**	**How**
1	2	see Overseer	shock reaction; back up
2	3	less scared (Overseer turns or goes away)	blend looks at Overseer and around world set Communicative Feature fear to maximum
3	2	more scared (Overseer turns or comes back)	quick jerk to Overseer; maybe back up
3	4	even less scared (Overseer has been far away for a while)	notice something interesting start looking at it
4	2	scared again (Overseer comes back)	scurry to back

FIGURE C.1: Mini-transitions that make up In Monitor

For **Explore World**:
Low-level signifiers:

1. Looking Around
2. Go to Spot
3. Look Around
4. Sigh
5. React to Overseer
6. Freeze in Place

Transitions:

From	**To**	**Reason**	**How**
1	2	Picked a spot that looked interesting or plausible	Focus on spot Look left, right Focus on spot again Go for it
1	5	Overseer came nearby	Whirl to face Overseer Back up Tremble
2	1	Overseer approached chosen spot (but not agent)	Shock reaction Watch Overseer Turn in opposite direction to pick something there
2	3	Got to spot	As approaches spot, look intently at object of interest
2	4	How sad! I miss the outside world!	Pause a moment in reflection
2	5	Overseer came nearby	Same as 1→5
2	6	I hallucinated the Overseer might be nearby	Glance around very quickly Turn and look at spot behind me.

FIGURE C.2: Mini-transitions that make up Explore World

From	**To**	**Reason**	**How**
3	1	Got bored.	Stare at object a moment Stare at feet Start looking around again
3	5	Overseer came nearby	Look at object Glance at Overseer Look at object Freeze
4	Any	Get your act together, little Patient	Stop and stare a moment Blink, blink Shake head while looking down Big sigh Back to work
5	1	Overseer went away again; The coast is clear	Watch Overseer leave Sigh Turn away from Overseer Squash down Sigh again Look over shoulder at Overseer Turn back away from Overseer
6	Any but 5	The coast is clear I just made it up	Look carefully around Shake head at folly Sigh
6	5	I'm paranoid, but I was right!	Same as 1-5

FIGURE C.3: Mini-transitions for Explore World, continued

For **Mope By Fence**:
Low-level signifiers:

1. Look Out At World
2. Sigh
3. Walk Up and Down Fence

Transitions:

From	To	Reason	How
1	2	Life is bad! Wish I was out there!	Stop looking a moment Lost in reverie
1	3	Bored with spot Get better position	Look in the direction I am planning to walk. Focus on something there Walk, keeping eye on spot
2	1	I'm sad, but I still want to look	Interruption
3	1	Got to point where I can see the thing I want to look at	Turn to face and look at the thing intently

FIGURE C.4: Mini-transitions that make up Mope By Fence

For **Read Sign**:
Low-level signifiers:

1. Read line
2. React to line

Transitions:

From	To	Reason	How
1	2	Saw something interesting	Shock reaction or re-read
2	1	Mulled it over	Pause Return to reading

FIGURE C.5: Mini-transitions that make up Read Sign

For **Exercise**:
Low-level signifiers:

1. Bob up and down

No Transitions.

FIGURE C.6: Mini-transitions that make up Exercise

For **Head-Banging**:
Low-level signifiers:

1. Hit Head on Ground
2. Wait to See if Light Went Out

Transitions:

From	**To**	**Reason**	**How**
1	2	Wants to get light on	
2	1	Light went out again	Act surprised Try to get light on by shaking head
2	1	Light went out again	Show frustration by freaking out

FIGURE C.7: Mini-transitions that make up Head-Banging

For **Unknown Behavior**:
Low-level signifiers:

1. Sigh
2. Look Around
3. Watch Overseer

Transitions:

From	To	Reason	How
1	2	Get your act together, little Patient	See sigh transition for Explore World
1	3	Sigh reminds you of your evil enemy	Turn slowly to Overseer Sigh again.
1	3	Just remembered you should be scared; or Overseer came nearby	Whirl around.
2	1	This place is bad	should work as interruption
2	3	Notice Overseer	Glance at Overseer Double-take If nearby, tremble and back up
3	1	What a pathetic piece of lamphood	Look away. Sigh.
3	2	Overseer went away	sigh and pause

FIGURE C.8: Mini-transitions that make up Unknown Behavior

For **Be Killed**:
Low-level signifiers:

1. Fear City
2. Die

Transitions:

From	To	Reason	How
1	2	Overseer hit button	Lightning bolt flash

FIGURE C.9: Mini-transitions that make up Be Killed

For **Turned Off**:
Low-level signifiers:

1. Be Turned Off

No Transitions.

FIGURE C.10: Mini-transitions that make up Turned Off

Composing High-Level Signifiers with Maxi-Transitions

Once each of the high-level signifiers was implemented, it was time to combine them with maxi-transitions to form the complete Behavior of the Patient. The design step for this is similar to that of composing the low-level signifiers. At this step, each possible transition between high-level signifiers is considered. For each possible transition, I listed the reasons for that behavioral change and corresponding ways to communicate that reason to the user. Figures C.11 through C.19 show the maxi-transition design for the Patient's high-level signifiers.

From **In Monitor:**

New Behavior	**Reason**	**How**
Explore world	Stops being so scared	Become curious. Move towards front. Look around carefully. Hop out. Still be a little scared for a while.
Head-banging	I am broken!	Here it is important to be clear as to what is going on. The Patient should look surprised, shake its head. Maybe the Overseer should look in disgust. When light goes on, Patient should be happy again.
Unknown behavior	Going on too long (?)	Pass in object of interest.

FIGURE C.11: Maxi-transitions from In Monitor

From **Explore World:**

New Behavior	**Reason**	**How**
In Monitor	Overseer comes right as Patient is coming out	Look at Overseer. Scurry back into monitor.
Read Sign	Notices schedule in its wandering.	Glance at schedule while walking by. Look interested. Walk over to it.
Mope by Fence	Gets near fence. Is bumming (after exercise). Oh, outside world! How cruel you are!! Wish I was back there.	Start slowing down beforehand. Life is bad. Look out at world. Sigh.
Head-Banging	If this happens, it's because the light goes out.	Look shocked. Look at camera so user can see your light is out. Shake your little head. Sideways, up and down. Swings get wider. Smack that head.
Turned Off	You've been moving around too much, getting too excited. Overseer doesn't like that	Maybe with your back to the Overseer, all of the sudden slump down. If you do see the Overseer, get scared. But keep moving so user sees the contrast. It's Overseer's job to make clear this is because of it.
Unknown Behavior	Explore world is going on a long time (?)	Pass in object of interest.

FIGURE C.12: Maxi-transitions from Explore World

From **Read Sign:**

New Behavior	**Reason**	**How**
Explore World	Overseer didn't bother it for some reason and it is done reading.	Stop reading. Stare at schedule. Sigh. Turn around. Look at world. Start exploring.
Exercise	Overseer comes over.	Sequence of looking at schedule and Overseer. Getting intimidated into it. Slow down as Overseer goes away.
Mope by Fence	Overseer is gone. Life is bad.	Slow down and stop. Sigh. Mope a little. Look at outside world. Sneak to the fence. Start moping.
Head-Banging	Light is out.	Make it short. Interruption.
Turned Off	Didn't pay attention that it was supposed to exercise	Like transition to exercise, but too sad to exercise. Sigh while looking at Overseer. Some pathetic attempts to exercise.

FIGURE C.13: Maxi-transitions from Read Sign

From **Exercise:**

New Behavior	**Reason**	**How**
Explore World Mope by Fence	Overseer is gone, it is bored and sad.	Must be sneaky. Slow down. Start looking around. Stop. Look at Overseer. Sneak off in other direction while keeping an eye on the Overseer.
Read Sign	Not done examining sign yet. Exercise is boring, sign is more interesting.	You're near the sign anyway (check). Turn around and start reading again. But slow down because you're not paying attention to exercise.
Head-Banging	Light goes out.	Quick interruption
Turned-Off	Not exercising enthusias-tically enough.	Exercise slowly. Don't notice Overseer coming. When Overseer comes near, exercise frantically, back up a little, but it's too late.
Unknown Behavior	I don't know	Object of interest

FIGURE C.14: Maxi-transitions from Exercise

From **Mope by Fence:**

New Behavior	**Reason**	**How**
Explore World	Bored of looking out of fence. Life must go on.	One last big sigh. Turn around. Scan Industrial Graveyard. Start exploring, but sadly.
Exercise	Supposed to be exercising. Overseer comes near.	Glance at Overseer. Turn back to fence. Slow exercises.
Head-Banging	Light goes out.	Look surprised (but resigned). Do a frustration dance.
Turned Off	Supposed to be exercising, but didn't notice Overseer.	Turn around at last second and cringe.
Unknown Behavior	I don't know	Object of interest

FIGURE C.15: Maxi-transitions from Mope by Fence

From **Head-Banging:**

New Behavior	**Reason**	**How**
Explore World Read Sign Exercise	Head-banging as interruption	If short: light goes back on right away, go back to activity.
Mope by Fence	Mope by fence is more serious	
Be Killed	Overseer noticed and is angry.	Uh-oh! When Overseer is near, start cringing. Look around, trying to figure out when Overseer is near, but can't see anything. Back up, maybe bumping into stuff.
Turned Off	Overseer saw and doesn't like it.	Patient doesn't notice Overseer coming. Just turn off (sudden break).
Unknown Behavior	?	?

FIGURE C.16: Maxi-transitions from Head-Banging

From **Be Killed**:
No transitions once you're dead.

FIGURE C.17: Transitions from Be Killed

From **Turned Off:**

New Behavior	**Reason**	**How**
Explore World	Turn off over	Slowly rise up. Shake self. Blink, blink. Maybe sigh. Look around slowly to get orientation. This should be exaggerated the first time, after that it becomes a routine.
Exercise	Same	Here you should be exercising like a maniac while looking around for the Overseer. Taper off.
Mope by Fence	Just another reason to be depressed	Same as first transition, but even more depressed.

FIGURE C.18: Maxi-transitions from Turned Off

From **Unknown Behavior:**

New Behavior	**Reason**	**How**
In Monitor	Overseer came near	Freak out and back up
Explore World	Bored of standing there	Fixate on a point; start walking towards there
Read Sign	You're near the sign anyway, and you haven't read it yet.	Glance at sign. Look with more interest. Start going.
Exercise	Overseer came near and it is time.	Look at Overseer. Look surprised. Go nuts.
Mope by Fence	Life is sad.	Sigh. Sweep your gaze across the inside of the junkyard. Look at the outside world. Then switch over whole-heartedly.
Head-banging	Light goes out.	Just like everyone else.
Turned off	Should be exercising.	See transition from mope by fence to head-banging.

FIGURE C.19: Maxi-transitions from Unknown Behavior

C.3 Complete Patient Design

Once these maxi-transitions are implemented, the Patient is complete. The full patient design is shown in Figure C.20. However, due to time constraints the entire design was not implemented. The design of the Patient as implemented is shown in Figure 7.54.

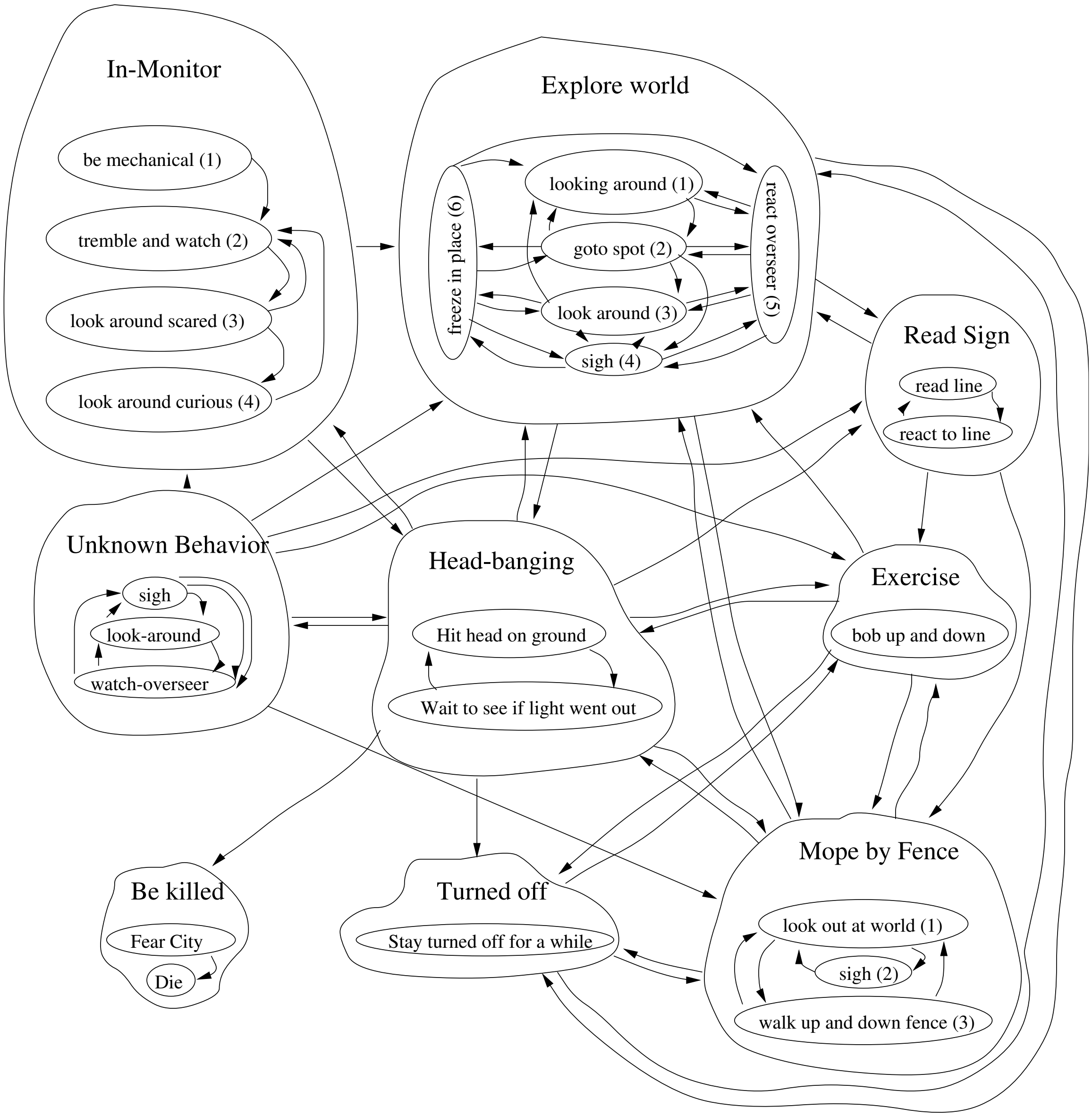


FIGURE C.20: The complete design of the Patient

Appendix D

Expostulations on Chapter 7 for the Technically Inclined

This Appendix consists of additions to Chapter 7 that are oriented for the reader whose technical interests are not exhausted by that populist rendering of the Expressivator. Pointers within the body of Chapter 7 will tell you when to read which part of this appendix.

D.1 Details on Transition Implementation

D.1.1 Transition Triggers

Transition triggers are complex sensors that look at conditions in the world to determine when it is time to switch from one behavior to another. Typically triggers test for things like the following:

- what behaviors are currently or have recently been run,
- what signs have recently been posted,
- events occuring in the virtual environment,
- communicative features,
- other transitions.

For example, when the Patient is hitting its head against the ground, it gets frustrated from time to time and switches from the "head-banging" low-level signifier to the "act frustrated" one. In order to determine when it is appropriate to switch, the transition trigger waits until the head-banging signifier has started running, and then counts the number of times a "smack head" sign has been posted, which corresponds to the number of times the user has seen the agent hit its head on the ground. After a sufficient number of smacks have occurred without the light going back on, the "act frustrated" transition trigger suggests to the rest of the system that it is time to change to the Patient's frustrated hopping-around behavior.

In general, when a trigger has found the right conditions for itself, it announces that fact to the rest of the system by finding the transition data structure associated with the signifier that called it, and modyifying it to reflect the trigger's opinion of what should be done. In particular, it notifies its parent signifier of which behavior should be terminated, which behavior should be started, and why.

D.1.2 Transition Demons

Transition demons keep an eye on the transition memory structures. They fire when an appropriate trigger has happened. Because it is generally more important to anticipate the new behavior properly than to finish up the old behavior in any particular way, transition demons generally check for transitions that are going to a particular behavior for a particular reason. Sometimes, they also check for the old behavior the agent was running.

The demon's job is to terminate the old behavior, go through a sequence of actions to create a transition, and then start the new behavior. The only exception is when the agent should merely interrupt a behavior, not terminate it; then the demon should make a transition, run the new behavior, and on termination make a transition back to the old behavior.

The transition demons' job is to kill the old behavior, do a transition sequence, and then start the new behavior. This sounds straightforward, but things are slightly more complicated. In particular, transitions must go from one behavior *that the user has seen* to another. For example, suppose the Patient has just decided to change from "look around scared" to "look around curiously." It has just killed off the "look around scared" behavior and is about to be curious when the Overseer approaches. Immediately, it is time for the Patient switch to "tremble and watch Overseer." Internally, this would mean a switch from "look around curious" to "tremble and watch Overseer" — but since the user does not know that the Patient is becoming curious, the correct transition is from "look around scared" to "tremble and watch Overseer." If this correctly chosen transition demon attempts to simple-mindedly kill the behavior from which it comes, "look around scared" (which no longer exists) will be killed and "look around curious" will continue on its merry way, running simultaneously with "tremble and watch Overseer." Oops.

To solve this kind of problem, transitions first delete, not just the old behavior they believe is running, but all other 'competing' behaviors. That is to say, mini-transitions kill any other low-level signifier that shares 'their' high-level signifier. Similarly, maxi-transitions kill any other high-level signifier. In order to make sure that now out-of-date transitions are deleted appropriately as well, mini-transitions are themselves declared as low-level signifiers, and maxi-transitions as high-level signifiers.

After transitions have killed preceding behaviors, they do some kind of transition sequence, and then start the requested new behavior. Mini-transitions add this to the high-level signifier that called them; maxi-transitions put it with the other high-level signifiers on the root of the agent's behavior tree.

If you have not yet had your fill of technical minutiae about transition implementation, I now refer you to section A.3 of the Appendix.

D.2 Technical Aspects to Expressivator Mindset Changes

The Expressivator was intended mainly as a way to add transitions to the basic Hap architecture, leaving the ordinary behavior structure alone. Nevertheless, it ends up fundamentally changing the meaning of both action-selection and behaviors in Hap.

D.2.1 Action-Expression in the Expressivator

Many agent architectures, especially those influenced by classical planning, require the agent designer to design behaviors based on their logical structure. For example, behaviors may be annotated with *preconditions* that state when they can be engaged in, and *postconditions* that note what changes they make to an environment. Action-selection then becomes a kind of problem-solving; you give the agent a goal to achieve in the world, and the agent chains behaviors until the last behavior's postcondition guarantees that the goal has been reached.

But there are many cases in which the 'point' of a behavior is not the changes the behavior may make in the environment, but the very behavior itself. 'Dancing,' for example, does not have any meaningful postconditions; the point of dancing is not to cause changes in the environment (unless it is a rain dance!), but for the pleasure of the activity itself[1]. The steps of the dance are not connected to one another by logical reasoning but by convention. There is, for example, no meaningful way for an agent to deduce that a foxtrot must consist of two long and two short steps; that's simply the way it's done. Many activities that are rooted in culture are similar. People usually do not stop for a rational analysis of when it is appropriate to say "hello," "thank you," or to ask someone how they are; they simply do it because it is conventional.

The action-selection mechanism in Hap is intended to reflect this concept of behaviors, not as means to achieve goals, but simply as sequences of actions to be engaged in for their own sake. Rather than having a designer specify the pre- and post-conditions for behaviors, both allowing and forcing the agent to reason about behavior before being able to act, the default in Hap is to have the designer specify behaviors as context-sensitive sequences of actions. The 'foxtrot' behavior will consist of the two long and two short steps simply because the designer wrote it that way; 'dancing' is done, not when the goal of dancing is achieved, but simply when the sequence of actions that make up dancing are complete.

[1] Of course, you can always call "pleasure" an effect on the world and measure the "pleasurableness" of various activities, and have the agent solve the problem of being happy by searching for and applying its maximally pleasurable activities. This neatly nails physical pleasure into the procrustrean bed of rationality by reducing temporally extended activity to a single step of goal satisfaction. There is little doubt that with sufficient ingenuity any activity can be mangled into goal-seeking rationality, but there should be at least a little scepticism about whether this is the best way of thinking about the problem.

In this framework, action-selection works largely by checking which behavior the agent is currently running and choosing the next step in the behavior. Behaviors are not simply scripted; they include annotations that let the agent know when the behavior is meaningful and when a running behavior no longer makes sense. However, by and large the reasoning behind the behavior's structure is implicit in the code for the behaviors as written by the designer.

Unlike Hap, the Expressivator demands that you know *why* your agent is doing what it does. The reasons for the agent's behavioral changes must be explicitly articulated in order to be expressed in transitions. Since transitions determine when and how it is appropriate to change from one signifier to another, they largely take over the role of action-selection for signifiers from the underlying architecture[2]. This means that, unlike Hap, explicit reasons for behavioral change form the basis for action-selection in the Expressivator.

At first, this change to Hap seemed unnatural: there did not seem to be any a priori reason why Hap action-selection should be inadequate for transitions. But the entire point of transitions is to show why you are switching from one behavior to another. If behaviors are simply sequenced, this means at some level you do not know why the behaviors are following one another; they simply do. That these reasons do not need to be articulated is an advantage in Hap because you do not always want to explain in fully logical, machine-understandable terms why the agent should do what it does. Nevertheless, it is a disadvantage if you want to express these reasons.

The Expressivator approach to action-selection is a compromise between the desire to include behavior whose logical structure cannot easily be elucidated, and the necessity to make reasons for behavioral choices explicit in order to express them. This is because the 'reasons' upon which the transitions are based need to be articulated *to the designer*, but not *to the machine*. Reasons for behavioral change are marked on transitions simply as tokens, such as "Patient-is-bored" or "Patient-saw-something-more-interesting." These reasons are not used by the agent to decide which activity makes sense, but by the designer as reminders of what the transition demon should express.

Still, the Expressivator does not include a full-fledged action-selection mechanism. For example, it could be that more than one transition triggers simultaneously, suggesting two conflicting behavioral changes. The Expressivator provides no mechanism to sort out which behavioral change should actually happen. I followed the style of Pengi, in making sure by hand that only one transition would ever fire in a particular circumstance. This strategy is not as painful as it sounds, because transitions are highly localized: (1) mini-transitions and maxi-transitions are handled separately; (2) mini-transitions can only ever conflict within their parent high-level signifier; (3) multiple transitions will only simultaneously fire when they are both transitions out of the same behavior. Nevertheless, it may be necessary in the future to add a full-fledged behavioral switching arbitration mechanism somewhat like that provided by Soar, which may check such things as the priorities of the various behaviors in question.

[2]Ordinary Hap action-selection still occurs for the subbehaviors which implement the signifiers

```
(parallel_production head_banging ()
  ;; initialize a pointer to myself
  (locals ("this_plan" "hh_plan_obj")
  ;; start my mini-transitions
  (with persistent
        (priority_modifier 200)
        (subgoal hit_head_to_wait_demon1 $$this_plan))
  (with persistent
        (priority_modifier 200)
        (subgoal hit_head_to_wait_demon2 $$this_plan))
  (with persistent
        (priority_modifier 200)
        (subgoal wait_to_hit_head_demon $$this_plan))
  (with persistent
        (priority_modifier 200)
        (subgoal freak_out_then_hit_head_demon
                 $$this_plan))
  ;; initialize my low-level signifiers
  (with (priority_modifier 100)
    (subgoal init_lls_headbanging))
  ;; start the first low-level signifier
  (with low_level_signifying
        (subgoal wait_for_light_on))
  ;; wait until the user notices me so I can
  ;; post myself
  (with effect_only
      (priority_modifier 300)
      (demon
        (("G (Goal CurrentLowLevelSignifier
                        == do_headbanging;);"))
        (post_high_level_signifier head_banging)))
  (wait))
```

FIGURE D.1: How Headbanging is invoked.

D.2.2 Behaviors in the Expressivator

High-level behaviors in Hap are simply some (context-sensitive) sequence of actions. In the Expressivator, on the other hand, a high-level signifier has a pre-given structure. Specifically, a high-level signifier consists simply of a set of low-level signifiers and the mini-transitions between them. When a high-level signifier is invoked, it simply starts a set of transition triggers and demons and the first low-level signifier (an example is in Figure D.1). After that, changes in low-level behaviors occur automatically as transitions trigger and then are implemented by transition demons.

Similarly, the full activity of the agent consists of the high-level signifiers and the maxi-transitions between them. When the agent starts up, it invokes all the maxi-transitions, and then starts the first high-level signifier. After that, the transitions take care of all subsequent changes to the agent's activity, triggering changes and modifying the agent's

The head-banging signifying behavior does the following things:

1. start its 4 mini-transitions
2. initialize its low-level signifiers
3. start the first low-level signifier
4. wait for the user to notice that it is happening, and then post itself to general memory

FIGURE D.2: Translation of previous figure for non-agent-builders and other interested parties.

behavioral structure as appropriate.

D.3 Behavior Transition Types

D.3.1 Explanatory Transition

The explanatory transition was the most useful, and I ended up using it for the majority of the transitions. They are easy to write — basically, you just make a short sequence of actions to explain what the agent is doing. Most of the time, they worked well. The only problem with explanatory transitions is that if you spend a lot of time in the explanatory sequence, the agent becomes less reactive. For example, the agent may be busy showing the user why it is about to read the schedule, and therefore not notice that the Overseer is about to attack. This problem can be ameliorated by varying the priority of various transitions, so that in this example the transition to reacting to the Overseer takes over even if the Patient is already in mid-transition. But in general, I found it was best to try to keep the transitions relatively short, if necessary by using meta-level controls to graft a transition-related activity onto the next behavior instead of doing it in the transition itself.

D.3.2 Subroutine Behavior Blend

A subroutine behavior blend involves combining two behaviors by adding a subroutine of the first behavior to the second behavior. For example, when the Patient goes from trembling at the Overseer to looking around scared, this is implemented by adding glances at the Overseer to look around scared. The subroutine behavior blend was easy to implement and did not require a lot of debugging. On the other hand, it was not so helpful from a narrative point of view; the behaviors probably would have made more sense with a clear, explained break between them.

The Mystery Transition

Relatively frequently, I would add a subbehavior to the new behavior, but it was not actually part of the old behavior, so it is not an ‘official’ subroutine behavior blend. For example, when the agent starts hitting its head in headbanging, the transition starts the headbanging behavior and then adds to it a subbehavior to first shake its head a few times to show the user its light has gone out and it is trying to get it back on again. This works nicely, though it could also be implemented as an explanatory transition. The main advantages over doing the additional subbehavior instead of an explanatory transition are (1) it can blend in with the other new behaviors' subbehaviors and (2) it makes sure the agent knows that this is "really" part of the second behavior, i.e. the current low-level signifier is set correctly as the new behavior instead of having the agent think it is in the nether region between the two behaviors.

D.3.3 Sudden Break

.

When used appropriately — i.e. not all the time, like in current architectures — this is both easy to do and very effective. The sudden break shows that the agent is having a visceral response to something going on around it. For example, when the Overseer comes near the patient, there is often a sudden break as the Patient whirls to face the Overseer and start trembling. Making this a sudden break makes it clear the Patient is not cogitating on the subject of the Overseer but rather having an immediate and intense reaction to it.

D.3.4 Interrupt

I use the interrupt-style transitions for behaviors that erupt during other behaviors. For example, the Patient may interrupt itself to sigh, and being turned off is also an interruption.

In general, I think the interrupt is dangerous. The turned off behavior, for example, can last a long time, and you probably don't want to return directly to the part of the behavior you were in last. For example, after being turned off by the Overseer and waking back up again, the Patient probably should not look intently at exactly the same spot on the trash in the world that it was looking at before.

This problem is compounded in Hap by the fact that behaviors don't really have any way of telling when they were interrupted (though signifiers could figure it out by seeing if they are still posted). This means after returning from an interrupt, a behavior may never realize anyone interrupted it; the behavior is completely oblivious to a fact that is essential to the user. In general, I think it would be better for behaviors like turned off that last a long time to kill the old signifier and start it all over again when they are done.

D.3.5 Reductive Behavior Blend

The reductive behavior blend reduces one of the behaviors to an attribute whose value can vary over time. The attribute is then applied to the other behavior. For example, when the Patient goes from looking around scared to looking around curiously, it first spends some time doing the scared version with fear set to a low value. Then, it goes to curious. This was easy to implement and blended the behaviors well: you could not tell when the change came between the scared behavior and the curious behavior. But for the same reason, this is a bad transition type from a narrative point of view: you do not know the agent is actually changing behaviors, or why the agent is becoming less scared. Again, a clear break with an explanation in between might have been more effective.

D.3.6 Off-screen Transition

I use this for almost all my transitions — the offscreen transition is built into the compiler. If the user is not looking at the agent, it immediately switches to the next behavior without a transition. This is useful in my system because transitions represent a kind of in-between state where the system is not totally sure which behavior it is in. It is therefore clearly best for the system to spend as little time in the transitions as possible.

This kind of transition might also be important in systems that have a function besides story or entertainment. In such a situation, it may be that transitions are for explanation, whereas the agent also has tasks to fulfill. In this case it's clearly best not to bother with explanation when the user is not paying attention.

For a few behaviors (for example, fear city to die) I left the transition in even when the behavior is not being watched. This was because the transition is so long that even if the user is not watching initially they may catch the end of the transition, and the transition is important enough to give the user the opportunity to see it. In some cases, I wait to change behaviors until I know the user is looking, so that s/he will not miss an important behavioral change.

D.3.7 Unknown Behavior

The unknown behavior is supposed to represent the default activity the agent does when it is not sure what to do. I wrote an Unknown Behavior for the patient, but I didn't end up using it in the system. If all your transitions are from and to a particular behavior, it doesn't make much sense to go to the unknown one for no reason. I also had a hard time coming up with good transitions for the Unknown Behavior since, by definition, you don't know why the agent is doing it. I therefore could not figure out how to get incorporate the unknown behavior in a logical way. It might be that in a different story — for example, where attention is not always focused on one agent — it may make more sense.

D.3.8 Principled Subroutine Behavior Blend

The idea of the principled subroutine behavior blend is to create a new behavior by combining already-running or new behaviors into a transition behavior. I use the principled subroutine behavior blend to go from being in the monitor to exploring the world. In this case, the Patient does a scared intermediate behavior that combines reacting to the overseer with stepping into the world while freezing in place at regular intervals.

This transition was difficult to write because it was basically like adding a whole new behavior. I could recycle some of the mini-transition demons but I also had to write some new ones specific to this maxi-transition. On the other hand, the behavior works well and is nicely reactive. In general, it is too much work, but it could be useful from time to time.

D.3.9 Symbolic Reduction

Under symbolic reduction, one behavior is reduced to a simple sign or symbol and incorporated into the other. I use this kind of transition when the Patient goes from reading the schedule to exercising. After launching the exercise behavior, I slowly reduce the energy as the Overseer goes away. This was very easy to write and works well. People definitely seem to understand what is going on.

D.3.10 Virtual Behavior Blend

In the virtual behavior blend, both behaviors run, but one of them has its muscle commands paralyzed. I use the virtual behavior blend when the Patient is turned off. This way, it would still have emotional reactions to the Overseer approaching, but would not actually move.

I found this kind of behavior blend exceptionally difficult to control. It had two major problems. Firstly, the agent would leap back into its old behavior the minute turned-off stopped paralyzing it, causing very strange behavioral discontinuities. Secondly, it was difficult to paralyze absolutely everything that needed paralyzing, with the result that the agent would still move around even though it was lying passed out on the ground. I fiddled with this transition extensively to get it right, but in the end, it did not seem to bring enough advantages to make it worth the effort.

D.4 Problems with Using Hap for Transitions

The number one problem with using Hap as a basis for the Expressivator is that you cannot pass around behavior names as Hap variables. Hap variables can only be integers, and for various reasons that have to do with the details of Hap's implementation in RAL it was not possible to encode goal names in a straightforward way as integers. The difficulty with this is that the transition system does some minimal reasoning about behaviors, and as soon as you start reasoning about them you need to be able to save them as variables. This would let you, for example, pass the behavior name to subbehaviors, save it in memory and call it later, and so on. Yes, it was always possible to find hacks around this problem, but this meant every instance of wanting to pass behavior names became an hour-long experiment in generating really horrific code.

This particular "feature" of Hap explains why I did write transitions that would go from any behavior to a particular behavior, but I never wrote transitions that went from a particular behavior to any other behavior. In order to do this, I would need to pass in to the generic transition the name of a behavior that it was going to have to start. But since I couldn't pass in the name of the behavior, this didn't happen. In general, I probably could have made a lot of the code much more general if I could have passed behavior names around.

Bibliography

[Agre and Chapman, 1987] Philip E. Agre and David Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, July 1987.

[Agre and Chapman, 1990] Philip E. Agre and David Chapman. What are plans for? In Pattie Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, pages 17–34. MIT Press, Cambridge, MA, 1990.

[Agre, 1988] Philip E. Agre. *The Dynamic Structure of Everyday Life*. PhD thesis, Massachusetts Institute of Technology Artificial Intelligence Laboratory, Cambridge, MA, 1988.

[Agre, 1990] Philip E. Agre. Portents of planning: A critical reading of the first paragraph of Miller, Galanter, and Pribram's 'Plans and the Structure of Behavior'. Paper presented at the Conference on Narrative in the Human Sciences, University of Iowa, July 1990., 1990.

[Agre, 1995] Philip E. Agre. The soul gained and lost: Artificial intelligence as a philosophical project. *Stanford Humanities Review*, 4(2), 1995. Special issue — Constructions of the Mind: Artificial Intelligence and the Humanities.

[Agre, 1997] Philip E. Agre. *Computation and Human Experience*. Cambridge University Press, Cambridge, UK, 1997.

[American Psychiatric Association, 1980] American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders (DSM-III)*. APA, Washington, D.C.:, 3rd edition, 1980.

[American Psychiatric Association, 1994] American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders (DSM-IV)*. APA, Washington, D.C.:, 4th edition, 1994.

[Analytix Inc., 1996] Analytix Inc. http://www.analytix.com/, 1996.

[Baker and Matlack, 1998] Stephen Baker and Carol Matlack. Chernobyl: If you can make it here... *Business Week*, pages 168–170, March 30 1998.

[Barthes, 1984] Roland Barthes. From work to text. In Brian Wallis, editor, *Art After Modernism: Rethinking Representation*, pages 169–174. New Museum of Contemporary Art, New York, 1984.

[Barton, 1995] Will Barton. Letting your self go: Hybrid intelligence, shared cognitive space and posthuman desire. Presented at Virtual Futures, Coventry, UK., 1995.

[Bates *et al.*, 1992] Joseph Bates, A. Bryan Loyall, and W. Scott Reilly. Integrating reactivity, goals, and emotion in a broad agent. Technical Report CMU-CS-92-142, Carnegie Mellon University, 1992. Also appeared in the Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society, Bloomington, Indiana, July 1992.

[Bates, 1994] Joseph Bates. The role of emotion in believable agents. Technical Report CMU-CS-94-136, Carnegie Mellon University, 1994. Also appears in Communications of the ACM, Special Issue on Agents, July 1994.

[Baur, 1991] Susan Baur. *The Dinosaur Man: Tales of Madness and Enchantment from the Back Ward*. Edward Burlingame Books, New York, 1991.

[Benyus, 1992] Janine M. Benyus. *Beastly Behaviors*. Addison-Wesley, Reading, MA, 1992.

[Blair and Meyer, 1997] David Blair and Tom Meyer. Tools for an interactive virtual cinema. In Robert Trappl and Paolo Petta, editors, *Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents*, number 1195 in Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, 1997.

[Blanchot, 1981] Maurice Blanchot. *The Madness of the Day*. Station Hill, Barrytown, New York, 1981. Trans. Lydia Davis.

[Bledsoe, 1986] Woody Bledsoe. I had a dream: AAAI presidential address. *AI Magazine*, 7(1):57–61, 1986.

[Blumberg and Galyean, 1995] Bruce Blumberg and Tinsley A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of SIGGraph*, 1995.

[Blumberg, 1994] Bruce Blumberg. Action-selection in Hamsterdam: Lessons from ethology. In *Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior*, Brighton, 1994.

[Blumberg, 1996] Bruce Blumberg. *Old Tricks, New Dogs: Ethology and Interactive Creatures*. PhD thesis, MIT Media Lab, Cambridge, MA, 1996.

[Brooks and Stein, 1993] Rodney A. Brooks and Lynn Andrea Stein. Building brains for bodies. Memo 1439, MIT AI Lab, August 1993.

[Brooks, 1986a] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23, April 1986.

[Brooks, 1986b] Rodney A. Brooks. Achieving Artificial Intelligence through building robots. Memo 899, MIT AI Lab, Month 1986.

[Brooks, 1990] Rodney A. Brooks. Elephants don't play chess. In Pattie Maes, editor, *Designing Autonomous Agents*. MIT Press, Cambridge, MA, 1990.

[Brooks, 1991a] Rodney Brooks. Integrated systems based on behaviors. In *Proceedings of AAAI Spring Symposium on Integrated Intelligent Architectures*, Stanford University, March 1991. Available in *SIGART Bulletin*, Volume 2, Number 4, August 1991.

[Brooks, 1991b] Rodney A. Brooks. Intelligence without reason. Technical Report AI Memo 1293, MIT AI Lab, 1991.

[Brooks, 1994] Rodney A. Brooks. Coherent behavior from many adaptive processes. In Dave Smith, editor, *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 22–29, Cambridge, MA, 1994. MIT Press.

[Brooks, 1995] Rodney A. Brooks. Intelligence without reason. In *The Artificial Life Route to Artificial Intelligence*, pages 25–81. Lawrence Erlbaum Associates, Hillsdale, NJ, 1995.

[Brooks, 1997] Rodney A. Brooks. From earwigs to humans. *Robotics and Autonomous Systems*, 20(2-4):291–304, June 1997.

[Bruner, 1986] Jerome Bruner. *Actual Minds, Possible Worlds*. Harvard University Press, Cambridge, MA, 1986.

[Bruner, 1990] Jerome Bruner. *Acts of Meaning*. Harvard University Press, Cambridge, MA, 1990.

[Bruner, 1991] Jerome Bruner. The narrative construction of reality. *Critical Inquiry*, 18(1):1–21, 1991.

[Carbonell, 1979] Jaime Carbonell. *Subjective Understanding: Computer Models of Belief Systems*. PhD thesis, Yale University Department of Computer Science, New Haven, CT, 1979.

[Chapman and Agre, 1986] David Chapman and Philip E. Agre. Abstract reasoning as emergent from concrete activity. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning about Actions and Plans*, pages 411–424. Morgan Kauffman, Los Altos, CA, 1986.

[Chapman, 1990] David Chapman. *Vision, Instruction, and Action*. PhD thesis, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1990.

[Cognitive Science, 1993] Cognitive science, January - March 1993. Volume 17. No. 1. Special Issue on Situated Action.

[Coleman *et al.*, 1984] James C. Coleman, James N. Butcher, and Robert C. Carson. *Abnormal Psychology and Modern Life*. Scott, Foresman and Co., Glenview, Illinois, 7 edition, 1984.

[Cooper, 1967] David Cooper. *Psychiatry and Anti-Psychiatry*. Ballantine Books, New York, NY, 1967.

[Crawford, 1993] T. Hugh Crawford. An interview with Bruno Latour. *Configurations*, 1(2):247–268, 1993.

[Crouch, 1990] Martha L. Crouch. Debating the responsibilities of plant scientists in the decade of the environment. *The Plant Cell*, pages 275–277, April 1990.

[Culler, 1982] Jonathan Culler. *On Deconstruction: Theory and Criticism after Structuralism*. Cornell University Press, Ithaca, NY, 1982.

[Curcuru,] Steve Curcuru. Personal Communication.

[Dautenhahn and Nehaniv, 1998] Kerstin Dautenhahn and Chrystopher Nehaniv. Artificial life and natural stories. In *International Symposium on Artificial Life and Robotics (AROB III)*, volume 2, pages 435–439, Beppu, Oita, Japan, 1998.

[Dawkins, 1989] Richard Dawkins. *The Selfish Gene*. Oxford University Press, Oxford, 1989.

[de Mul, 1997] Jos de Mul. Networked identities: Human identity in the digital era. In Michael B. Roetto, editor, *Proceedings of the Seventh International Symposium on Electronic Art*, pages 11–16, Rotterdam, 1997.

[DeLanda, 1991] Manuel DeLanda. *War in the Age of Intelligent Machines*. Zone Books, NY, 1991.

[Deleuze and Guattari, 1977] Gilles Deleuze and Félix Guattari. *Anti-Oedipus: Capitalism and Schizophrenia*. Viking Press, NY, 1977. Translated by Mark Seem.

[Deleuze and Guattari, 1987] Gilles Deleuze and Félix Guattari. November 28, 1947: How do you make yourself a body without organs. In *A Thousand Plateaus: Capitalism and Schizophrenia*, chapter 6, pages 149–166. University of Minnesota Press, Minneapolis, 1987. Translated by Brian Massumi.

[Dennett, 1987] Daniel Dennett. *The Intentional Stance*. MIT Press, Cambridge, MA, 1987.

[Derrida, 1976] Jacques Derrida. *Of Grammatology*. Johns Hopkins University Press, Baltimore, 1976. Translated by Gayatri Chakravorty Spivak.

[Don, 1990] Abbe Don. Narrative and the interface. In Brenda Laurel, editor, *The Art of Human-Computer Interface Design*, pages 383–391. Addison-Wesley, Reading, MA, 1990.

[Doray, 1988] Bernard Doray. *From Taylorism to Fordism: A Rational Madness*. Free Association, London, 1988. Trans. David Macey.

[Elliott *et al.*, 1998] Clark Elliott, Jacek Brzezinski, Sanjay Sheth, and Robert Salvatoriello. Story-morphing in the affective reasoning paradigm: Generating stories semi-automatically for use with 'emotionally intelligent' multimedia agents. In Katia P. Sycara and Michael Wooldridge, editors, *Proceedings of the Second International Conference on Autonomous Agents*, New York, May 1998. ACM Press.

[Foerst, 1996] Anne Foerst. Artificial Intelligence: Walking the boundary. *Zygon, Journal for Religion and Science*, 31:681–693, 1996.

[Foerst, 1998] Anne Foerst. Cog, a humanoid robot, and the question of the image of god. *Zygon, Journal for Religion and Science*, 33:91–111, 1998.

[Foner, 1993] Lenny Foner. What's an agent, anyway? http://foner.www.media.mit.edu/people/foner/Julia/Julia.html, 1993. Published in a revised version in *The Proceedings of the First International Conference on Autonomous Agents (AA '97)*.

[Forgy, 1991] C. L. Forgy. *Rule-extended Algorithmic Language Language Guide*. Production Systems Technologies, Inc., 1991.

[Foucault, 1973] Michel Foucault. *Madness and Civilization*. Vintage Books, New York, 1973. Trans. Richard Howard.

[Frank *et al.*, 1997] Adam Frank, Andrew Stern, and Ben Resner. Socially intelligent virtual petz. In Kerstin Dautenhahn, editor, *Proceedings of the 1997 AAAI Fall Symposium on Socially Intelligent Agents*, Menlo Park, CA, November 1997. AAAI Press. AAAI Technical Report FS-97-02.

[Fujita and Kageyama, 1997] Masahiro Fujita and Koji Kageyama. An open architecture for robot entertainment. In W. Lewis Johnson, editor, *Proceedings of the First International Conference on Autonomous Agents*, pages 435–442, New York, February 1997. ACM Press.

[Gadamer, 1986] Hans-Georg Gadamer. *Truth and Method*. Crossroad, NY, 1986.

[Galyean, 1995] Tinsley Galyean. *Narrative Guidance of Interactivity*. PhD thesis, MIT Media Lab, June 1995.

[Georgeff and Ingrand, 1989] M.P. Georgeff and F.F. Ingrand. Decision-making in an embedded reasoning system. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 972–978, Detroit, MI, 1989.

[Goffman, 1961] Erving Goffman. *Asylums: Essays on the Social Situaiton of Mental Patients and Other Inmates*. Anchor Books, Garden City, NY, 1961.

[Goldstein, 1995] Kurt Goldstein. *The Organism: A Holistic Approach to Biology Derived from Pathological Data in Man*. Zone Books, NY, 1995.

[Gove, 1986] Philip Babcock Gove, editor. *Webster's Third New International Dictionary of the English Language Unabridged*. Merriam-Webster, Springfield, MA, 1986.

[Gross and Levitt, 1994] Paul R. Gross and Normal Levitt. *Higher Superstitions: The Academic Left and Its Quarrels with Science*. Johns Hopkins University Press, Baltimore, 1994.

[Guattari, 1984] Félix Guattari. *Molecular Revolution*. Penguin Books, New York, 1984. Trans. Rosemary Sheed.

[Haraway, 1990a] Donna Haraway. A cyborg manifesto: Science, technology, and socialist-feminism in the late twentieth century. In *Simians, Cyborgs, and Women: The Re-Invention of Nature*, pages 149–181. Free Association, London, 1990.

[Haraway, 1990b] Donna Haraway. Situated knowledges: The science question in feminism and the privilege of partial perspective. In *Simians, Cyborgs, and Women: The Re-Invention of Nature*, pages 183–201. Free Association, London, 1990.

[Harding, 1994] Sandra Harding. Is science multicultural? challenges, resources, opportunities, uncertainties. *Configurations*, 2(2):301–330, 1994.

[Hayes *et al.*, 1994] Patrick J. Hayes, Kenneth M. Ford, and Neil Agnew. On babies and bathwater: A cautionary tale. *AI Magazine*, 15(4):15–26, 1994.

[Hayles,] N. Katherine Hayles. How we became posthuman: Virtual bodies in cybernetics, literature, and informatics. Unpublished Manuscript.

[Hayles, 1993] N. Katherine Hayles. The materiality of informatics. *Configurations*, 1(1):147–170, 1993.

[Hayles, 1997] N. Katherine Hayles. Narrative and the question of embodiment in scientific inquiry. Society for Literature and Science, 1997.

[Horswill, 1993] Ian Horswill. Polly: A vision-based artificial agent. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 824–829, Menlo Park, July 1993. AAAI Press.

[Hsu *et al.*, 1990] Feng-hsiung Hsu, Thomas S. Anantharaman, Murray S. Campbell, and Andreas Nowatzyk. Deep thought. In T. Anthony Marsland and Jonathan Schaeffer, editors, *Computers, Chess, and Cognition*. Springer Verlag, New York, 1990.

[James, 1998] Ellen James. *Her Protector*. Harlequin Superromance. Harlequin Books, Toronto, 1998.

[Janet, 1889] Pierre Janet. *L'Automatisme Psychologique: Essai de Psychologie Expérimentale sur les Formes Inférieures de l'Activité Humaine*. Ancienne Librairie Germer Baillière et Cie, Paris, 1889. Ed. Félix Alcan.

[Joachims *et al.*, 1997] Thorsten Joachims, Dayne Freitag, and Tom Mitchell. WebWatcher: A tour guide for the World Wide Web. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, August 1997.

[Johnson, 1997] W. Lewis Johnson, editor. *Proceedings of the First International Conference on Autonomous Agents*, NY, February 1997. ACM Press.

[Josephson and Josephson, 1962] Eric Josephson and Mary Josephson. Introduction. In Eric Josephson and Mary Josephson, editors, *Man Alone: Alienation in Modern Society*, pages 9–53. Dell Publishing, NY, 1962.

[Keller, 1985] Evelyn Fox Keller. *Reflections on Gender and Science*. Yale University Press, New Haven, 1985.

[Kennedy, 1989] Noah Kennedy. *The Industrialization of Intelligence: Mind and Machine in the Modern Age*. Unwin Hyman, Boston, 1989.

[Kirk and Kutchins, 1992] Stuart A. Kirk and Herb Kutchins. *The Selling of DSM: The Rhetoric of Science in Psychiatry*. A. de Gruyter, New York, 1992.

[Laing and Esterson, 1970] R. D. Laing and A. Esterson. *Sanity, Madness, and the Family*. Penguin Books, Ltd., Middlesex, England, 1970.

[Laing, 1960] R. D. Laing. *The Divided Self: An Existential Study in Sanity and Madness*. Penguin Books, Middlesex, England, 1960.

[Laird, 1991] John Laird, editor. *Proceedings of AAAI Spring Symposium on Integrated Intelligent Architectures*, March 1991. Available in *SIGART Bulletin*, Volume 2, Number 4, August 1991.

[Lanier, 1996] Jaron Lanier. My problem with agents. *Wired*, 4(11), November 1996.

[Lasseter, 1987] John Lasseter. Principles of traditional animation applied to 3D computer animation. In *Proceedings of SIGGRAPH '87*, pages 35–44. Association for Computing Machinery, 1987.

[Laurel, 1986] Brenda Laurel. Interface as mimesis. In *User-Centered System Design*, pages 67–85. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[Laurel, 1991] Brenda Laurel. *Computers As Theatre*. Addison-Wesley, Reading, MA, 1991.

[Lecocq, 1991] Sylvain Lecocq. Untitled (letter to his doctor). In John G. H. Oakes, editor, *In the Realms of the Unreal: "Insane" Writings*, pages 158–161. Four Walls Eight Windows, New York, 1991. Trans. Roger Cardinal.

[Lester and Stone, 1997] James C. Lester and Brian A. Stone. Increasing believability in animated pedagogical agents. In W. Lewis Johnson, editor, *Proceedings of the First International Conference on Autonomous Agents*, pages 16–21, NY, February 1997. ACM Press.

[Lester *et al.*, 1997] James C. Lester, Sharolyn A. Converse, Susan E. Kahler, S. Todd Barlow, Brian A. Stone, and Ravinder S. Bhogal. The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of CHI '97*, pages 359–366, Atlanta, March 1997.

[Levins and Lewontin, 1985] Richard Levins and Richard C. Lewontin. *The Dialectical Biologist*. Harvard University Press, Cambridge, MA, 1985.

[Lewontin *et al.*, 1984] Richard C. Lewontin, Steven Rose, and Leon J. Kamin. *Not In Our Genes: Biology, Ideology, and Human Nature*. Pantheon Books, New York, 1984.

[Lewontin, 1991] R. C. Lewontin. *Biology as Ideology: The Doctrine of DNA*. Harper Perennial, New York, 1991.

[Lewontin, 1995] Richard C. Lewontin. À la recherche du temps perdu. *Configurations*, 3(2):257–265, 1995.

[Loyall,] A. Bryan Loyall. Personal communication.

[Loyall and Bates, 1991] A. Bryan Loyall and Joseph Bates. Hap: A reactive, adaptive architecture for agents. Technical Report CMU-CS-91-147, Carnegie Mellon University, 1991.

[Loyall and Bates, 1993] A. Bryan Loyall and Joseph Bates. Real-time control of animated broad agents. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, Boulder, Colorado, June 1993.

[Loyall, 1997a] A. Bryan Loyall. *Believable Agents: Building Interactive Personalities*. PhD thesis, Carnegie Mellon University, Pittsburgh, May 1997. CMU-CS-97-123.

[Loyall, 1997b] A. Bryan Loyall. Personality-rich believable agents that use language. In W. Lewis Johnson, editor, *Proceedings of the First International Conference on Autonomous Agents*, NY, February 1997. ACM Press.

[Lukács, 1971] Georg Lukács. Reification and the consciousness of the proletariat. In *History and Class Consciousness: Studies in Marxist Dialectics*. MIT Press, Cambridge, MA, 1971. Trans. Rodney Livingstone.

[Maes, 1989a] Pattie Maes. The dynamics of action-selection. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, volume 2, pages 991–997, Detroit, MI, August 1989. Morgan Kaufmann.

[Maes, 1989b] Pattie Maes. How to do the right thing. AI Memo 1180, MIT AI Laboratory, December 1989.

[Maes, 1990a] Pattie Maes, editor. *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back.* MIT Press, Cambridge, MA, 1990.

[Maes, 1990b] Pattie Maes. Guest editorial: Designing autonomous agents. In Pattie Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, pages 1–2. MIT Press, Cambridge, MA, 1990.

[Maes, 1990c] Pattie Maes. Situated agents can have goals. In Pattie Maes, editor, *Designing Autonomous Agents*, pages 49–70. MIT Press, Cambridge, MA, 1990.

[Maes, 1991] Pattie Maes. Behavior-based Artificial Intelligence. In Jean-Arcady Meyer and Stewart W. Wilson, editors, *From Animals to Animats 2*, pages 2–10, Cambridge, MA, 1991. MIT Press.

[Maes, 1993 1994] Pattie Maes. Modeling adaptive autonomous agents. *Artificial Life*, 1(1-2):135–162, Fall-Winter 1993-1994.

[Mahoney, 1980] Michael S. Mahoney. Reading a machine. In N. Metropolis, J. Howlett, and G.-C. Rota, editors, *A History of Computing in the Twentieth Century: A Collection of Essays*, pages 3–9. Academic Press, NY, 1980.

[Mahoney, 1997] Michael S. Mahoney. Software and the assembly line. Carnegie Mellon Software Engineering Institute Invited Talk, February 1997.

[Marx, 1967] Karl Marx. *Capital: A Critique of Political Economy*, volume I: "The Process of Capitalist Production". International Publishers, New York, 1967. Ed. Frederick Engels. Trans. Samuel Moore and Edward Areling.

[Massumi, 1992] Brian Massumi. *A User's Guide to Capitalism and Schizophrenia: Deviations from Deleuze and Guattari*. MIT Press, Cambridge, MA, 1992.

[Mateas, 1997] Michael Mateas. Computational subjectivity in virtual world avatars. In Kerstin Dautenhahn, editor, *Proceedings of AAAI-97 Workshop on Socially Intelligent Agents*, pages 87–92, 1997. Available from AAAI as Technical Report FS-97-02.

[McDermott, 1981] Drew McDermott. Artificial Intelligence meets natural stupidity. In John Haugeland, editor, *Mind Design: Philosophy, Psychology, Artificial Intelligence*, chapter 5, pages 143–160. MIT Press, Cambridge, MA, 1981.

[Mérő, 1990] László Mérő. *Ways of Thinking: The Limits of Rational Thought and Artificial Intelligence*. World Scientific, New Jersey, 1990. Edited by Viktor Mészáros. Translated by Anna C. Gősi-Greguss.

[Miedaner, 1981] Terrel Miedaner. The soul of the Mark III beast. In Douglas R. Hofstadter and Daniel C. Dennett, editors, *The Mind's I: Reflections on Self and Soul*, pages 109–115. Basic Books, Inc., New York, 1981.

[Minsky, 1988] Marvin Minsky. *The Society of Mind*. Simon and Schuster, New York, 1988.

[Minuchin *et al.*, 1978] Salvador Minuchin, Bernice L. Rosman, and Lester Baker. *Psychosomatic Families: Anorexia Nervosa in Context*. Harvard University Press, Cambridge, MA, 1978.

[Mumford, 1934] Lewis Mumford. *Technics and Civilization*. Harcourt, Brace, and Company, New York, 1934.

[Neal Reilly, 1996] Scott Neal Reilly. *Believable Social and Emotional Agents*. PhD thesis, Carnegie Mellon University, 1996. CMU-CS-96-138.

[Newell and Simon, 1972] Allen Newell and Herbert A. Simon. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ, 1972.

[Newell, 1981] Allen Newell. The knowledge level. Technical Report CMU-CS-81-131, Carnegie Mellon University Department of Computer Science, 1981.

[Newell, 1990] Allen Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1990.

[Nilsson, 1984] Nils J. Nilsson. Shakey the robot. Technical Report 323, AI Center, SRI International, Menlo Park, CA, 1984.

[Noble, 1998] David F. Noble. Digital diploma mills: The automation of higher education. *First Monday*, 3(1), January 1998. http://firstmonday.dk/issues/issue3_1/noble/index.html.

[Norman, 1993] Donald A. Norman. Cognition in the head and in the world: An introduction to the special issue on situated action. *Cognitive Science*, 17:1–6, 1993.

[O'Neill-Brown, 1997] Patricia O'Neill-Brown. Setting the stage for the culturally adaptive agent. In *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents*, pages 93–97, Menlo Park, CA, 1997. AAAI Press.

[Payton *et al.*, 1992] David W. Payton, David Keirsey, Dan M. Kimble, Jimmy Krozel, and J. Kenneth Rosenblatt. Do whatever works: A robust approach to fault-tolerant autonomous control. *Journal of Applied Intelligence*, 2:225–250, 1992.

[Pearson *et al.*, 1993] Douglas J. Pearson, Scott B. Huffman, Mark B. Willis, John E. Laird, and Randolph M. Jones. Intelligent multi-level control in a highly reactive domain. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, Pittsburgh, PA, 1993.

[Penny, 1995] Simon Penny. Living machines. *Scientific American*, September 1995. 150th Anniversary Issue.

[Penny, 1997a] Simon Penny. Embodied cultural agents at the intersection of robotics, cognitive science, and interactive art. In Kerstin Dautenhahn, editor, *Socially Intelligent Agents: Papers from the 1997 Fall Symposium*, pages 103–105, Menlo Park, 1997. AAAI Press. Technical Report FS-97-02.

[Penny, 1997b] Simon Penny. The virtualisation of artistic practice: Body knowledge and the engineering world view. *CAA Art Journal*, Fall 1997. Ed. Johanna Drucker. Special Issue on Electronic Art.

[Perlin, 1995] Ken Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15, March 1995.

[Petta and Trappl, 1997] Paolo Petta and Robert Trappl. Personalities for synthetic actors: Current issues and some perspectives. In Paolo Petta and Robert Trappl, editors, *Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents*, number 1195 in

Lecture Notes in Artificial Intelligence, pages 209–218. Springer Verlag, Berlin, 1997.

[Pixar, 1986] Pixar. Luxo, jr. (film), 1986.

[Porter, 1997] Tom Porter. Depicting perception, thought, and action in Toy Story. In *First International Conference on Autonomous Agents*, February 1997. Invited Talk.

[Rheingold, 1995] Howard Rheingold. Technology criticism, ethics, and you. *San Francisco Examiner*, November 1 1995. Also available in ZK Proceedings: Net Criticism, Volume 1, p. 110, http://www.desk.nl/ nettime/zkp/.

[Rhodes, 1996] Bradley James Rhodes. PHISH-nets: Planning heuristically in situated hybrid networks. Master's thesis, MIT Media Lab, 1996.

[Ritzer, 1993] George Ritzer. *The McDonaldization of Society: An Investigation into the Changing Character of Contemporary Social Life*. Pine Forge Press, Newbury Park, CA, 1993.

[Robear Jr., 1991] James Walter Robear Jr. Reality check. In John G. H. Oakes, editor, *In the Realms of the Unreal: "Insane" Writings*, pages 18–19. Four Walls Eight Windows, New York, 1991.

[Ronell, 1989] Avital Ronell. *The Telephone Book: Technology — Schizophrenia — Electric Speech*. University of Nebraska Press, Lincoln, 1989.

[Rouse, 1993] Joseph Rouse. What are cultural studies of science? *Configurations*, 1(1):57–94, 1993.

[Sacks, 1995] Oliver Sacks. Forward to Kurt Goldstein's *The Organism: A Holistic Approach to Biology Derived from Pathological Data in Man*. Zone Books, NY, 1995.

[Samuel, 1995] Arthur L. Samuel. Some studies in machine learning using the game of checkers. In Edward A. Feigenbaum, editor, *Computer and Thought*. AAAI Press, Menlo Park, 1995.

[Schempf, 1995] Hagen Schempf. BOA: Asbestos pipe-insulation removal robot sydtem. Technical Report CMU-RI-TR-95-15, Carnegie Mellon University Robotics Institute, Pittsburgh, PA, 1995.

[Sengers,] Nishka Sengers. Personal Communication.

[Sengers, 1998] Phoebe Sengers. Do the thing right: An architecture for action-expression. In *Proceedings of the Second International Conference on Autonomous Agents*, May 1998. To appear.

[Shakes *et al.*, 1997] Jonathan Shakes, Marc Langheinrich, and Oren Etzioni. Dynamic reference sifting: A case study in the homepage domain. In *Proceedings of the Sixth International World Wide Web Conference*, pages 189–200, 1997.

[Shklovsky, 1990] Victor Shklovsky. *Theory of Prose*. Dalkey Archive Press, Elmwood, Ill, 1990. Trans. Benjamin Sher.

[Simmons *et al.*, 1997] Reid Simmons, Richard Goodwin, Karen Zita Haigh, Sven Koenig, and Joseph O'Sullivan. A modular architecture for office delivery robots. In W. Lewis Johnson, editor, *Proceedings of the First International Conference on Autonomous Agents*, pages 245–252, NY, February 1997. ACM Press.

[Smithers, 1992] Tim Smithers. Taking eliminative materialism seriously: A methodology for autonomous systems research. In Francisco J. Varela and Paul Bourgine, editors, *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 31–47, Cambridge, MA, 1992. MIT Press.

[Snow, 1969] C. P. Snow. *The Two Cultures and the Scientific Revolution*. Cambridge University Press, London, 1969.

[Social Text, 1996] Social Text, April 1996. Volume 14, Nos. 1-2. Ed. Andrew Ross. Special Issue on the Science Wars.

[Sokal, 1996] Alan Sokal. A physicist experiments with cultural studies. *Lingua Franca*, pages 62–64, May-June 1996.

[Sontag, 1979] Susan Sontag. *Ilness as Metaphor*. Vintage Books, New York, NY, 1979.

[Steels and Brooks, 1995] Luc Steels and Rodney Brooks, editors. *The Artificial Life Route to Artificial Intelligence*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1995.

[Steels, 1994] Luc Steels. The Artificial Life roots of Artificial Intelligence. *Artificial Life*, 1(1-2):75–110, 1994.

[Stone, 1996] Brian A. Stone. Dynamically sequencing an animated pedagogical agent. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 424–431, Portland, OR, August 1996.

[Strasser, 1982] Susan Strasser. *Never Done: A History of American Housework*. Pantheon Books, New York, 1982.

[Suchman, 1987] Lucy A. Suchman. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, 1987.

[Swados, 1962] Harvey Swados. The myth of the happy worker. In Eric and Mary Josephson, editors, *Man Alone: Alienation in Modern Society*, pages 106–113. Dell Publishing, NY, 1962.

[Sycara and Wooldridge, 1998] Katia P. Sycara and Michael Wooldridge, editors. *Proceedings of the Second International Conference on Autonomous Agents*, NY, May 1998. ACM Press.

[Thomas and Johnston, 1981] Frank Thomas and Ollie Johnston. *Disney Animation: The Illusion of Life*. Abbeville Press, New York, 1981.

[Tyrell, 1993] Toby Tyrell. *Computational Mechanisms for Action Selection*. PhD thesis, University of Edinburgh, 1993.

[Van den Haag, 1962] Ernest Van den Haag. Of happiness and of despair we have no measure. In Eric and Mary Josephson, editors, *Man Alone: Alienation in Modern Society*, pages 180–199. Dell Publishing, NY, 1962.

[Varela *et al.*, 1991] Francisco J. Varela, Evan Thompson, and Eleanor Rosch. *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press, Cambridge, MA, 1991.

[Various, 1985] Various. *Die Bibel*. Deutsche Bibelgesellschaft, Stuttgart, 1985. Translated by Martin Luther.

[Vere and Bickmore, 1990] Steven Vere and Timothy Bickmore. A basic agent. *Computational Intelligence*, 6:41–60, 1990.

[Viscott, 1972] David S. Viscott. *The Making of a Psychiatrist*. Arbor House, New York, NY, 1972.

[Walter, 1963] W. Grey Walter. *The Living Brain*. W.W. Norton, 1963.

[Washington, 1991] Karoselle Washington. The killing floors. In John G. H. Oakes, editor, *In the Realms of the Unreal: "Insane" Writings*, pages 48–52. Four Walls Eight Windows, New York, 1991.

[Watt, 1993] Alan Watt. *3D Computer Graphics*, chapter 13. Addison-Wesley, Reading, MA, 2nd edition, 1993.

[Wavish and Graham, 1996] Peter Wavish and Michael Graham. A situated action approach to implementing characters in computer games. *AAI*, 10, 1996.

[Webb *et al.*, 1981] Linda J. Webb, Carlo C. DiClemente, Ediwn E. Johnstone, Joyce L. Sanders, and Robin A. Perley, editors. *DSM-III Training Guide for Use with the American Psychiatric Association's Diagnostic and Statistical Manual of Mental Disorders (Third Edition)*. Brunner / Mazel, New York, 1981.

[Weizenbaum, 1965] Joseph Weizenbaum. Eliza — a computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery*, (1):36–45, 1965.

[Wernecke, 1994] Josie Wernecke. *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor (TM), Release 2*. Addison-Wesley, Reading, MA, 1994.

[Winograd and Flores, 1986] Terry Winograd and Carlos F. Flores. *Understanding Computers and Cognition: A New Foundation for Design*. Ablex Pub. Corp., Norwood, NJ, 1986.

[Winograd, 1972] Terry Winograd. *Understanding Natural Language*. Academic Press, New York, 1972.

[Wise, 1996] J. MacGregor Wise. Intelligent agency. Society for Literature and Science, 1996.