Observation-based Expectation Generation and Response for Behavior-based Artificial Creatures

by

Christopher John Kline
B.S., Computer Science
College of Engineering
Cornell University, Ithaca, NY
May 1997

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES
at the
Massachusetts Institute of Technology
September 1999

Signature of Author ____
Program in Media Arts and Sciences
August 6, 1999

Certified by ____
Bruce M. Blumberg
Asahi Broadcasting Corporation Assistant Professor of Media Arts and Sciences
MIT Media Laboratory
Thesis Supervisor

Accepted by ____
Stephen A. Benton
Chairperson, Departmental Committee on Graduate Students
Program in Media Arts and Sciences
MIT Media Laboratory

Observation-based Expectation Generation and Response for Behavior-based Artificial Creatures

by

Christopher John Kline

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on August 6, 1999
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES

Abstract

This thesis seeks to address the incorporation of a low-level cognitive ability into reactive, behavior-based artificial intelligence architectures. Specifically, it addresses the need to generate short-term, observation-based expectations about the world and react appropriately to the violation of those expectations. In it I discuss the motivation for incorporating expectations into a reactive behavior-based architecture, outline the qualitative properties of expectations and the conditions under which they may be violated, propose a model for generating expectations and responding to their violation, detail one implementation of such a model, and finally propose this work as a starting point from which future work on higher-order cognition and behavior might begin.

Thesis Supervisor: Bruce M. Blumberg, MIT Media Laboratory
Title: Asahi Broadcasting Corporation Assistant Professor of Media Arts and Sciences

Observation-based Expectation Generation and Response for Behavior-based Artificial Creatures

by

Christopher John Kline

The following people served as readers for this thesis:

Reader: ______________________________

Aaron F. Bobick
Associate Professor of the College of Computing
Georgia Institute of Technology

Reader: ______________________________

Marvin L. Minsky
Professor of Electrical Engineering and Computer Science
Toshiba Professor of Media Arts and Sciences Emeritus
Massachusetts Institute of Technology

Acknowledgments

My advisor, Bruce Blumberg, provided invaluable advice, encouragement, and friendship throughout my time at MIT. In return, I taught him one very important lesson: never, *ever* disable *all* of the processors on a multiprocessor machine.

I feel privileged to have had the opportunity to work with such a talented and fun team for the past two years. For their inspiration, friendship, and hard work I would like to thank the Synthetic Characters group: Marc Downie, Michal Hlavac, Michael P. Johnson, Delphine Nain, Kenneth Russell, Dan Stiehl, Bill Tomlinson, Jed Wahl, and Song-Yee Yoon.

Sumit Basu, Yuri Ivanov, Tony Jebara, Andrew Wilson, Chris Wren, and the rest of the Vismod crew have my gratitude for helping with all things statistical. Andrew Wilson has my special thanks for sharing his friendship, great advice, and thousands of cups of coffee.

Whitman Richards saw an early version of this work and challenged me to formalize some of my ideas. This forced me to rewrite nearly everything from scratch, but results are the better for his suggestions. Aaron Bobick, in addition to being one of the readers for this thesis, also gave me many helpful suggestions in the early stages of development.

I would also like to acknowledge my parents, Ronald and Carole, and my sister Melinda, for twenty-five years of constant love and support.

And to all of my friends, past and present—thank you. You know who you are.

Contents

List of Figures

Chapter 1

Introduction

This thesis seeks to address the addition of a low-level cognitive ability into an architecture for building reactive, behavior-based artificial creatures. Specifically, it addresses the need to generate short-term, observation-based expectations about the world and react appropriately to the violation of those expectations.

This work has been informed by the study of ethology, cognitive science, and other fields, as well as a great deal of introspection into my own behavior and that of the animals I observe on an everyday basis. As such, I would be quite pleased if its major concepts are or have been born out by rigorous cognitive or ethological studies. However, I would like to stress that the ideas proposed herein are *not* an attempt at a cognitive theory of expectations in humans or other *real* animals. Rather, they are attempt at providing a foundation for constructing *synthetic* creatures whose behavior seems reasonable, explicable, and believable in the eyes of *human* observers.

In the following sections of this chapter I will motivate this work by explaining how it relates to problems extant in behavior-based architectures and then introduce the expectation-based approach with which I have addressed these problems. In subsequent chapters I will propose a model for the generating expectations and responding appropriately to their violation, detail one implementation of such a model, and finally discuss some ways in which this work might serve as a starting point from which future work on higher-order cognition and behavior might begin.

1.1 Motivation

Humans expect a certain degree of sophistication in the behavior of real animals, and therefore if our goal is to build an artificial animal whose behavior seems plausible and life-like in the eyes of human observers, that animal should be capable of such sophistication.

To date, behavior-based artificial intelligence research has approached this problem by focusing on the issues of adequacy, relevance, and coherence [Bro91a]. Adequacy ensures that a creature's behavioral repertoire is sufficient for achieving the creature's goals. Given that adequate behaviors exist, relevance involves achieving the appropriate balance between the competing influences of internal motivations and external stimuli such that the creature chooses the most appropriate behavior. Often more than one behavior is appropriate at any given time, so the creature must not only choose the correct action at the correct time, but also take care that behaviors exhibit the right amount of persistence and do not interfere with each other or hinder progress towards the goal by alternating too rapidly; this is known as coherence.

Though behavior-based AI has yielded impressive results in action-selection simulations and utilitarian robotics, attempts at building creatures that are meant to seem like real animals—three-dimensional, embodied creatures exhibiting complex, coherent and relevant behavior in "real-time" over extended periods—reveal that these issues are far from having been resolved. The root of these unresolved problems, as detailed in the following sections, lies in attempting to use reactive behavior-based architectures to build these creatures without giving them even the most basic type of action-oriented reasoning that real animals use to compensate for the limitations of perception.

1.1.1 The problem of perception

From the standpoint of behaviorists, animal behavior does not require cognition at all; instead, complex behavior is treated as a property emerging from the activity of many simple behaviors whose control structure is tightly coupled to a dynamic external environment.

In some cases this makes sense. As Agre and Chapman point out [AC87], much of everyday behavior is spent in generic "routines": notice an obstacle, step around it; hear a sound, turn to look; taste something bad, spit it out. See, react, repeat. In these cases, because the behavior is so directly and reflexively coupled to the stimulus, the world is often "its own best model" [Bro91b] and reasoning is not necessary, efficient, or even desirable.

But if you are simply reacting to what you observe, what happens when you cannot make an observation? Noises are fleeting, odors blow away in the breeze, and objects in the world are constantly moving in and out of occlusion. Perception is noisy, and any creature with a plausible model of perception must continually make decisions based on incomplete information about its current situation.

To illustrate why realistic perception is a problem for behavior-based AI, imagine a behavior-based creature in the form of a hungry cat that is chasing an agile mouse. As the mouse dodges left and right, the cat will continually adjust its motion ("see, react, repeat") to move in the direction of the prey. But the instant the mouse jumps behind a wall, the cat will stop the chase and begin the next most appropriate activity, such as sitting down to rest. Why would the cat engage in such unnatural behavior when the mouse was "obviously" only a short distance away? Because our feline friend has no understanding of its environment beyond the moment-to-moment stimuli it receives from its senses—from the cat's point of view, the mouse no longer exists! A similar thing would happen if the mouse ran in and out of the cat's field of view, causing the cat to spasmodically jump up and sit down over and over again as if it were in the throws of a neurological malfunction.

This problem inevitably crops up in stateless reactive behavior-based systems because they have no way of coping with the fleeting nature of real-world perception. Every moment is startlingly new because they have no concept of the past or the future. Because they do not have the capability for even the simplest forms of action-oriented reasoning, they cannot achieve the robustness and complexity of real organisms.

1.1.2 The need for assumptions

As we have just seen, the world can only be its own model when it is directly perceivable, and this presents a serious impediment to building robust creatures. Take a moment to contemplate your own everyday behavior. When you wake up in the morning, how do you find the bathroom? While driving, how do you avoid hitting the cars around even though you can only catch fleeting glimpses of their movement? When you chase someone who runs behind a wall, how do you know to catch him when he comes out the other side?

The answer is that we work from assumptions, extrapolating from observations made in the past to arrive at a "best guess" of the present and future. You assume that the bathroom in your house is down the hallway on the left because you have no reason to assume otherwise; that is where it was every time you had ever seen it. The cars were moving parallel to you when you last saw them, so you assume that they kept traveling in the same general direction. The same holds for chasing someone behind a wall.

This kind of basic "common sense" reasoning is incredibly important because all but the simplest of intelligent organisms base the vast majority of their daily routines upon observations that either happened in the past or have yet to occur. Any creature that needs to function in a dynamic world with limited perception must make assumptions in order to achieve behavior that is adequate, relevant, and coherent.

1.1.3 Expectations and the importance of being wrong

An important thing to note is that, from the standpoint of building artificial animals, the limitations of perception should not be looked upon as a curse or flaw. These limitations are part of what makes animals behave in an "animal-like" way. An omniscient creature would not behave like a real animal at all, because real animals are not omniscient. The right approach is to strive for an understanding of how real animals cope with these limitations, and then integrate these coping mechanisms into our architectures. One way in which we will know that we have succeeded is that our synthetic animals will make the same mistakes as real ones.

But before discussing mistakes I need to talk a bit about how expectations fit into the picture. Expectations differ slightly from assumptions, and I differentiate them by saying that expectations are assumptions about the future state of the world. So why are expectations important? One reason is that real animals spend a lot of time dealing with the future. When deciding among several adequate behaviors in which to engage, the most relevant behavior is often the one that has the highest chance of being successful or being maximally efficient. For example, when crossing the street, I could either wait for all the cars to pass, or I could begin walking immediately under the assumption that, if my expectations prove correct, I will finish crossing before the cars are anywhere near me.

As essential as expectations are to making decisions, it is equally if not more important for life-like behavior that a creature respond in a believable manner when those expectations are not met; cognitive scientists call these failure conditions *expectation violations*. If, while crossing the street, a car suddenly appears closer to me than I had anticipated, I should be surprised. If a cat chases a mouse behind a wall but the mouse is nowhere to be seen when the cat goes behind to take a look, the cat will be confused.

The expression of an expectation violation is one of the important clues humans use to gain insight into what humans and other animals are thinking, and we have come to expect characteristic responses to common types of violations such as confusion, surprise, and disbelief. In fact, because humans project their own cognitive processes onto the animals they observe, if a synthetic animal does *not* respond in the expected way to a typical violation scenario then the illusion of life is ruined. That in itself is an important reason to incorporate violations into any model of reasoning. But expectation violations do more than change the message animals convey to observers—they also have an important effect on a creature's behavior. The more our expectations fail to match the eventual reality and the greater the discrepancy between the two (i.e., the less predictable something becomes), the less we trust in our assumptions, changing our behavioral choices.

So now let us get back to the point I skipped over earlier. Both classical and behavior-based AI have spent years learning how to build systems that know how to "do the right thing" based on their internal motivations and the state of the world around them. Historically, classical AI systems have been criticized for being competent in narrow and well-defined domains but somewhat confounded by quickly-changing environments. On the other hand, behavior-based AI has had success at creating systems which are good at adjusting to unpredictable environmental changes, but these systems seem "short-sighted" when pursuing goals and experience coherency problems due to perceptual restrictions. These problems are well known, and there have been attempts to build architectures which incorporate the best of both worlds [Fir87, Mae90].

With respect to building believable, life-like creatures with which humans can empathize, past work in both fields of AI has failed to identify the single most important reason for incorporating an expectation mechanism: *without expectations, animals cannot make mistakes*. So why in the world would we want to try building a creature that makes mistakes if we can't even build one that can "do the right thing"? Because much of what makes animals "animals-like" are the mistakes they make.

Think of a puppy ... part of what makes playing with a puppy so compelling is that the puppy is capable of being confused, misled, surprised, and teased. Why else would humans spend hours watching a dog sprint after a ball that they had only pretended to throw?

Why else would deception play such a major role in the behavior of real animals unless real animals were gullible?

In this respect expectations are the key because they provide a legitimate grounding (or perhaps "excuse") for the behavior of the creature. One of the reasons why artificial intelligences built to date seem so inept and unnatural is because the "mistakes" they make are the kind of errors that no real animal would ever make[1]. That is, animals do not make arbitrary mistakes—real animals only do the "wrong" thing, speaking objectively, when it is done for the "right" reason[2]. These true mistakes are the result of choosing a course of action based upon a set of incorrect assumptions about the present or future condition of those parts of the world which cannot presently be directly perceived.

1.2 Proposed solution

I hope that I have made a strong case for the necessity of expectations and expectation violations when building life-like synthetic animals. Now, so to speak, I need to sell you on the product.

I propose that the first step, overcoming the limitations of perception, can be achieved by giving the characters the ability to use assumptions to fill in gaps in observational data, thereby reducing the behavioral discontinuities characteristic of existing behavior-based AI architectures. This means endowing them with an understanding of object persistence—the notion that objects in the world are separate entities that continue to exist when not directly perceivable—which developmental psychologist Jean Piaget has theorized to be the foundation of intelligence [Pia52, Pia54]. The various components implicitly necessary for object persistence will be discussed, such as temporally-based assumptions, assumption confidences, and sensory grounding of observations.

However, object persistence in itself is meaningless without a concept of expectations—after all, what does it mean to "understand that a hidden object still exists" without the expectation that the object will still be there when the occluder is removed? Therefore,

[1] For this reason I believe that a reverse Turing test—trying to appear more machine-like to a human than an actual machine—would be as difficult for a human to pass as the original test has been for machines.

[2] This is not to say that animals use the same reasoning processes as humans. In fact, real animals often lack certain reasoning abilities that most humans would consider trivial. Krushinskii's [Kru62] well-documented (though relatively unknown) experiments on expectations (which he, being a behaviorist, called "extrapolation reflexes") in pigeons, ducks, fowls, crows, and rabbits show a remarkable and often amusing variation in the level of reasoning sophistication in each species.

as a second step, I propose a mechanism by which the aforementioned components form the foundation of a notion of expectations and the conditions under which they may be violated, and discuss how these violations lead to changes in the behavior of the associated creature. These expectations in effect bootstrap the creature with *some* of the reasoning abilities[3] associated with the traditional Piagetian developmental stages, and allow it to overcome many of the problems identified discussed in this chapter[4].

I also hope to show that the addition of expectations is not wholly at odds with the behavior-based approach. Essentially, my approach allows behavior-based creatures to react to two separate but causally-related worlds—the perceived state of the world and the assumed state of the world.

Some might call this "imagination".

[3] I am referring to the ability of a creature to reason about the autonomous behavior of objects, as opposed to reasoning about the effects of its actions upon those objects. For a very thorough attempt at computationally reverse-engineering all aspects of Piagetian development, and an excellent discussion of Piagetian theory in general, I point the reader to the work of Gary Drescher [Dre91].

[4] At first glance it may seem that I am attempting to add some form of classical AI "capital-P" Planning capability to a reactive system, but this is not the case. My approach to expectations and the use of pre- and post-conditions in Planning differs as follows: Planning involves using assumptions about how the state of the world will change as a result of the creature's actions upon it, whereas my expectations are predictions of the future state of the world without *any* intervention by the creature. The approaches also differ in the way expectations are generated (top-down versus bottom-up); this will be discussed in Section 4.

My expectations avoid the "frame problem" [MH69]—understanding what aspects of a situation will *not* change as a result of a particular action or event—because they do not attempt to predict how the actions of the creature will affect the future, nor do they attempt to account for the implications of one expectation upon another. Though this might cause seizures in more than one AI researcher, the idea is "give me just enough reasoning so that I can act intelligently when I have incomplete information, and if anything more complex comes up my reactive behavior will handle it."

Chapter 2

A Theory of Expectations and Expectation Violations

The chapter seeks to answer the question "what are expectations and expectation violations" in an intuitive yet somewhat more formal manner than that of Chapter 1. The intent is to provide the reader with a description of the qualitative behavior of expectations that is adequate enough to begin construction of such a system.

2.1 Assumption model

For the purposes of this work, an assumption is defined as a triple

$$A \ = < S(O),\, c,\, t >$$

defining the state S of an object O at some time t, hypothesized with confidence c. The confidence is a continuous value in the range $[0, 1]$. When discussing time I will designated the present as $t = t_{now}$.

A state is a collection of n features

$$S = \{\, f_0,\, f_1,\, \ldots\,,\, f_{n-2},\, f_{n-1}\,\}$$

where a feature is some observable property of an object. Each feature f_i may in itself be a state; in this way object states can be hierarchically organized. For example, my current state might consist of three features: my name, my position, and my clothing, the last of which consists of two features—my shirt and my pants—which are further defined by their color and cleanliness.

2.1.1 Properties of assumptions

The purpose of these assumptions is to enable creatures to function properly in the absence of complete information. As such, it is advantageous from the standpoints of plausibility and design that assumptions exhibit two fundamental properties:

1. **transparency:** an assumption should be indistinguishable, in terms of structure and method of manipulation, to an equivalent real-world observation made by the creature's sensory apparatus

The principle behind transparency is that reasoning with assumptions should utilize the same processes as reasoning from observations. In this sense I am presuming that real animals have not evolved completely separate mechanisms for reasoning with assumptions; while objectively the animal may have some notion that these assumptions are not "real", in the majority of instances this should not make a difference. Additionally, this symmetry makes it easy for creature designers to understand the role of assumptions and how they might be incorporated into an expectation architecture.

2. **saliency:** an assumption should have the capacity to elicit an appropriate behavioral response; in the absence of any disparaging information about the likelihood of a particular assumption (i.e., $c = 1$), the assumption should be capable of eliciting the same behavior response as the equivalent physical percept

Whereas transparency ensures that assumptions are capable of being processed by the same machinery as observations, saliency insures that assumptions and observations are evaluated in the same "common currency" when making behavioral decisions. For example, if I assume that there is a particularly tempting piece of food under the box in front of me, yet my thesis is due in two hours, the decision of whether to eat or to write should be as difficult as if the food were plainly visible.

2.1.2 Observations, assumptions, and confidence

Perception is only an approximation of reality—our eyes are capable of detecting only discrete differences in a very small band of all incoming radiation, the spatial resolution of touch is limited by the distribution of exteroceptors, sensing in general is temporally limited to the response time of individual neurons, etc. When one "sees a slug", for example, what has actually occurred is that a noisy measurement was made of the state of an object which, given the value of its features, fit the concept of a slug. That measurement was made at a specific time and the observer now has some confidence in the accuracy of that measurement. So in some sense reasoning is based *completely* upon assumptions.

In my model an assumption is created immediately upon the first observation of an object; its timestamp is set to the time of observation, its features are initialized to the value of the corresponding measurements, and the assumption is assigned an initial confidence[1]. This confidence is the primary means by which the saliency of an assumption is determined. For example, when $t = t_{now}$ and $c = 1$ the creature has complete faith that the assumption is an accurate reflection of the current state of the corresponding object. Sometimes an observation is less trustworthy, such when one catches a brief glimpse of a housefly. In this case the confidence would be lower and the distinction between assumptions and reality is more explicit; rather than saying "it was a fly" one might say "I had the *impression* that it was a fly" or "I *assume* that it was a fly".

In this system there are three fundamental ways in which the confidence in an assumption varies over time according to the availability and nature of subsequent measurements:

1. **Temporal variability.** Confidence should rise over time so long as there are new measurements available; similarly, it should decline while there are no measurements to reinforce the assumption.

   A good illustration of temporal variability is the sighting of a ghost. The initial sighting of the ghost generates the assumption that there is a spooky translucent spectre floating in front of you; though skeptical at first, your confidence grows the longer you examine it. Were it to disappear, however, the strength of your conviction would rapidly decline.

[1] The initial confidence is influenced by a variety of factors, some of which might include top-down "hints" derived from on past experience, the current emotional state of the creature, the creature's focus of attention at the time of the observation, etc.

2. **Temporally normalized decay.** Suppose your only observations of the ghost's state occur when you see it float past a doorway. Furthermore, suppose the duration during which observations can be made is very brief (0.5 seconds) and the time between them quite long (60 minutes). In this scenario your confidence should decay quickly between each observation. However, if the ghost moved past the doorway slowly and often, so that observations were long (10 seconds) with little delay in between, then your confidence should remain high throughout the encounter because it would decay slowly between observations.

   Temporally normalized decay means that the rate at which a confidence decays is related to the ratio of the combined observation time to the total amount of time that the observed object is presumed to have existed.

   More formally,

$$\frac{dc}{dt} \sim \left(\frac{\int_{t_0}^{t_f} f(t)\, dt}{t_f - t_0} - 1 \right) \tag{2.1}$$

$$f(t) = \sum_{t_i} \delta(t - t_i)$$

   where t_0 is the time of the initial observation, t_f is the time of the final observation, $\delta(t)$ is the impulse function, and each t_i is a time for which a measurement was available.

3. **Grounding errors.** Imagine that the ghost floats through the doorway and is hidden from your sight. On one hand, you cannot see the ghost anymore, so you should be less confident that it is actually in the next room. On the other hand, the fact that you cannot make additional observations is perfectly reasonable since you cannot see through walls. But your confidence should certainly take a rapid plunge if the ghost disappears before your eyes, because "things just don't disappear".

   This requires a notion of *sensory grounding*, which answers the question "given my understanding of the way the world works and my assumption of the object's state, does it 'make sense' that I can (or cannot) currently make a measurement of the object's state?" Another way of looking at this is to say that "unless something inexplicable has happened, either I can make a measurement of the object's state or I know why I cannot make such a measurement".

The basis for this grounding is a combination of one or more sensory/cognitive systems such as vision or touch. Though it may be augmented by later experience (and I will discuss this in Section 2.2.2), what I am referring to here is the kind of extremely low-level "common sense" understanding about one's self and the world that is either innate or acquired during the very early stages of development.

You can think of sensory grounding as a function of two boolean inputs, the first of which states whether or not a measurement *can* be made while the second indicates if that measurement *should* be able to be made. The function returns a boolean result indicating either acceptance or rejection of the perceived state of the world, as shown in Figure 2-1. The two accept cases indicate normal situations such as when your car is in front of you and you are able to determine its color (`accept-2`), or when your car is parked around the corner and you cannot see it (`accept-1`). The reject cases occur when something inexplicable has happened, such as if your car suddenly disappeared without a trace while you were staring at it (`reject-1`), or if there was a giant pink elephant[2] in your bathtub one morning (`reject-2`). These reject cases are called *grounding errors*.

The fact that confidence decays rapidly in the face of grounding errors is something I like to call the "mirage effect". A traveler spots what appears to be an oasis while wandering aimlessly through the scorching desert. Over the course of an hour he wanders towards it, watching the camels bend their heads to drink from a pool of water and anticipating the shade of the palm trees. Given the continuous visual contact over a long period of time, the traveler has every reason to believe that this vision is in fact a reality. But as he reaches down to draw water from the pool, the touch of his hand upon dry earth instantly discredits the oasis and his confidence

[2] It should be noted that the theory presented in this document do not yet properly handle the `reject-2` condition wherein a measurement can be made of something for which measurement should be impossible. It is capable of detecting the simple case of when you can see something which, given your assumption, should not be visible. However, the pink elephant situation described above requires a higher level of reasoning, implying that the observer can *quickly* determine that a given percept is either 1) possible or 2) impossible in the current context, both of which are problems of equally mind-boggling complexity. Regardless, successful organisms seemingly make use of the latter test, which Minsky [Min86] refers to as negative knowledge, on an everyday basis.

		Should you be able to make a measurement?	
Can you make a measurement?		No	Yes
	No	`accept-1`	`reject-1`
	Yes	`reject-2`	`accept-2`

Figure 2-1: The sensory grounding function

rapidly disappears.

2.1.3 Assumptions about the present are not enough

In the previous section I identified the core properties of assumptions, transparency and saliency, and discussed the basic principles behind changes in assumption confidence; I hope that those properties and principles now seem intuitively obvious to the reader. Unfortunately, though building assumptions may help a creature deal with gaps in perceptual data, the theory described so far is inadequate for building artificial creatures with the level of robust and life-like behavior that real animals possess.

First, the astute reader may have noticed that I have avoided any mention of how an assumption's state, $S(O)$, changes with each new observation. $S(O)$ is an approximation of the actual state of object O, and in theory the accuracy of this approximation should improve with each new measurement and degrade with each missing one. Using the housefly example of Section 2.1.2, the first observation of the fly might allow only a poor estimate to be made of its position, but each successive observation should improve that estimate. In practice the integration of new observations might be implemented in any number of ways, some examples of which will be discussed in Chapter 3.

Second, in order to build creatures capable of having reactions such as surprise and confusion, they must fundamentally have some way of knowing when their assumptions do not agree with reality. This can be done in three ways: 1) by making an assumption about the present and then evaluating it at some point in the future ("hindsight"); 2) by comparing an assumption about the present with an observation of the present ("direct observation"); 3) by making an assumption about some point in time in the future and then evaluating that assumption through observation when the time arrives ("prediction"). In a reactive system hindsight is not useful because reactive creatures never reflect upon the past. The second approach is invalidated on the grounds that, if direct observation were possible, the creature would not need to use an assumption. This leaves prediction as only usable means for evaluating the accuracy of assumptions in a reactive system.

Third, not only must a creature be able to compare an assumption with reality, it must also know whether or not the difference between them "makes sense". In this regard basic assumptions are inadequate because they incorporate only a static model of observational plausibility (through sensory grounding). The ghost of Section 2.1.2 provides a good example—the sighting of a ghost is plausible if the assumed position of the ghost is non-occluded; in the event that the ghost is not visible at the assumed position, the lack of an observation is plausible only when the position is occluded. In other words, either you *should* be able to see the ghost or you *should not*, and those conditions are fixed. But when you spend a lot of time around ghosts you begin to learn that they have a habit of disappearing in front of your eyes. Given this new knowledge, why would you be surprised if it happened again?

How do you decide whether to believe your senses or to ignore your senses and trust your assumption? Strangely enough, the answer often lies in the assumption itself! For example, if a desk were to disappear before your eyes you would be quite surprised[3], not only because your senses are telling you that this is implausible, but also because in all your experience you have never seen a desk do that. But, assuming that you eventually came to accept this strange behavior of desks, you will be less prone to surprise the next time it happens. In this way past history is important as sensory grounding for assessing the plausibility of observations[4].

What lies at the bottom of these problems is an understanding of the relationship between the past, present, and future. In the next section I overcome this limitation by

[3]It is interesting to note that being surprised when something disappears requires a different path of reasoning than being surprised when something appears unexpectedly, because in the latter case there is no prior assumption and the creature must have had an awareness of what was *not* part of the greater context just prior to the appearance.

[4]Many thanks to my advisor, Bruce Blumberg, for pointing this out.

introducing a special kind of assumption that I call an *expectation*.

2.2 Expectation model

At the syntactic level an expectation is not much different from a basic assumption: an expectation E is defined as an assumption with the restriction that $t = t_{now} + \delta$ for $\delta > 0$, meaning that it is an assumption about the future state of an object O. But what it implies is that the creature is not only approximating $S(O)$ from noisy sensor data, but also making a *prediction* of that state at some time in the future. For behavior-based architectures this means that the creature is no longer reacting to the present; it is instead living in the near future.

In a dynamic environment this policy of staying "one step ahead" of the world around you makes a great deal of sense. The smaller the value of δ the closer the prediction is to the current actual state of the object, and the accuracy of the expectation improves with an increase in the rate of expectation updates. Therefore in the best case the prediction is 1) not very different from the present, and 2) pretty close to the future. In terms of behavior, this gives the creature the ability react appropriately while still being able to to anticipate reasonable changes in the environment.

More importantly, because the creature is generating beliefs about the future, it now has some metric by which to identify when something unexpected has happened.

2.2.1 Expectation violations

The reaction to an unusual observation is known as an *expectation violation* (EV). For future clarity, I will define an *unexpected* observation as one which is potential cause for an expectation violation.

From the standpoint of building believable artificial animals, violations are important because they let the observer know that the animal is intelligent—a reaction to something unusual shows that the animal has an certain level of awareness about the kind of behavior that is to be expected from objects in the world[5]. From the purposes of facilitating robust behavior, EVs are a good indication that the current level of confidence in (and thus the

[5]Cognitive psychologists use these reactions as indicators of expectation violations, thereby being able to test animals for particular classes of expectations.

salience of) an assumption is unwarranted and should be changed.

An unexpected observation can occur in two situations, the first of which is when an observation causes a grounding error. An example of this would be watching a ball roll behind one end of a wall but failing to observe it roll out the other side. The second occurs when the difference between the expected value of a feature and the measured value is highly unlikely, like if your supervisor walked into work naked one morning.

In the latter case there are two implicit prerequisites. First is the classic correspondence problem of how to associate a prior observation with a new one. Intuitively, this boils down to a way in which the creature can answer questions like "is this red ball the same red ball I saw before?" One way to deal with this is to essentially ignore the problem and assume that it is handled at a very low level by the underlying architecture; for example, by comparing object pointers in a software implementation or relying on a hardware primitive to provide a unique identifier for each object. A more psychologically-plausible approach would be to perform a rough comparison between the features in the expected and observed states. A significant difference between the two, or a grounding error resulting from the assumption that the observation corresponds to the expectation, would provide a strong indication that the observed object is not the one corresponding to the expectation.

Second, there needs to be some definition of what it means for the discrepancy between the prediction and the observation to be "unlikely". In practice this could be implemented in a myriad of different ways; if feature values were modeled probabilistically a good measure might be how many standard deviations out the observation was from the predicted mean ("how did that object get all the way over there?"). For boolean features any change at all might be cause for violation ("well, he used to have a head, but now it's gone!"). Another approach would be to look for discontinuities as the value of the feature changes. It is my suspicion that in real animals this is implemented in a very messy, efficient, feature- and task-specific manner.

Chapter 3 talks about how these two requirements are handled in my implementation; for now, let us assume that they exist and are somehow available to the underlying implementation (be it biological or algorithmic).

Assuming that the creature has experienced an unexpected observation, the first thing to do is determine whether or not that observation warrants a violation and, if so, what type of violation has occurred. In this work I have concentrated on three types of violations:

confusion, surprise, and disbelief.

Confusion

Confusion occurs whenever an observation is unexpected and the creature is at least minimally confident in the expectation. It is the most frequently encountered expectation violation because the conditions under which it occurs are the same as those commonly found in the initial stages of learning; i.e., little experience and inconsistent success.

One scenario for confusion would be a creature observing an object whose rapid motion it has so far been unable to predict in a reliable fashion. Another would be the previous example of seeing a ball roll behind a wall but not observing it roll out the other side.

Surprise

If an observation is physically available, surprise requires a medium level of confidence in the expectation. In effect, the creature is saying "I am fairly confident that I understand the behavior of the observed object; therefore what it just did is either unnatural or the result of some ability that was previously unknown to me."

However, if the observation is implausibly missing the creature must have a high level of confidence before surprise will occur. The reason for the high confidence requirement is because confusion is more appropriate under conditions of low confidence. Intuitively, it is the difference between "I guess I didn't understand that object as well as I thought" and "something weird has occurred, because that object should have been visible."

Disbelief

Disbelief occurs whenever there is an *extremely* high level of confidence associated with an unexpected violation. In this case the creature is so confident in its model of the object that it is unable to accept the observation (either the measured value or the lack of a measurement) as being based in reality. Discovering five billion dollars in your normally subsistence-level bank account is a perfect candidate for a disbelief violation, as is walking to work and finding that your office building has vanished without a trace.

Disbelief is somewhat unique in how it influences the way that the model underlying the expectation is updated; this is discussed in Section 2.2.3.

2.2.2 How violations affect behavior

We mentioned earlier that expectation violations affect the behavior of a creature in two ways: first by influencing the confidence in an expectation, and second by potentially triggering behaviors indicative of the violations. In the following two sections I elaborate on each of these in turn.

Violations and salience

Imagine that you are building an artificial dog that lives in an artificial house, and that you have built into your dog a desire to eat on a regular basis. Perhaps there are several bowls of food distributed around the house, all of which the dog has come into contact with with varying frequency in the past. So ... when your dog wakes up famished from an afternoon nap, where should he look for food? The kitchen is very close, but it has been almost a day since he last visited that area, so there is a fair chance that the bowl there may be empty. On the other hand, your dog saw an open bag of dog food near the back door earlier this morning; unfortunately, reaching the back door requires descending two flights of stairs.

This illustrates how expectations influence the behavioral decision-making process. Whereas before the only factors in the "common currency" of behavior-based AI were internal motivations and sensory stimuli, creatures are now able to balance long-term goals and short-term opportunistic behavior through a form of cost-benefit analysis based on expectation confidence[6].

As your dog wanders around the house he is constantly updating the confidence in his expectation of whether or not each bowl contains food. As he observes each bowl the confidence in the corresponding expectation rises; in between observations his confidence decays because he cannot be sure how the bowl's state has changed. This is the principle of temporal variability discussed in Section 2.1.2. Until now, however, the *rate* at which the confidence changes has not been discussed.

When deciding on a policy for determining an appropriate rate of change in confidence there several factors to consider. First, when observations are unavailable or have cause violations, the confidence should decrease at rate commensurate with the probability that the expectation will eventually be borne out. The confidence in a passing observation is likely

[6]An obvious analog to this is the concept of a "discounting factor" commonly used in machine learning techniques

to decay much more quickly than something a creature experiences on a continuing basis, for example, because the latter example has been reinforced by a history of interactions.

Likewise, the confidence's rate of restitution should reflect the creature's faith that the model underlying the expectation is a good one. This means that confidence should recover slowly when the expectation has been undependable in the past—the "once bitten, twice shy" principle—and it should also bounce back quickly after a temporary setback in otherwise good track record.

What we really need is a notion of *reliability*, meaning an estimate of how good the underlying expectation model is at predicting the future state of the object. Given a reliability $0 \leq r \leq 1$ and a confidence c:

$$c_t = c_{t-1} + k * r * dt \tag{2.2}$$

where

$$k = \begin{cases} -\alpha & \text{no violation; measurement available,} \\ -\beta & \text{no violation; measurement unavailable,} \\ +\gamma & \text{violation; measurement available,} \\ +\eta & \text{violation; measurement unavailable} \end{cases} \tag{2.3}$$

α, β, γ, and η are scaling factors controlling the relative conservatism of the creature under various conditions.

It is important to note that a decrease in reliability is not justified simply because an expectation is not met. If I expect my mother to be wearing a blue blouse when I come home to visit, but instead she is wearing a red blouse, is the underlying expectation model less reliable? Probably not, because the likelihood of the observation was within some epsilon of acceptability. But if she were wearing an African dashiki or something equally out of character, then perhaps I should adjust my model.

One way to accommodate this is the notion of an expectation-based reliability metric:

$$r_e = \frac{(T - V)}{T} \tag{2.4}$$

where T and V are the total number of times the expectation for this object has been updated and the number of updates for which a violation occurred, respectively[7]. Note that T includes the updates in which a measurement could not be made.

The expectation-based reliability concept is useful because it achieves the objectives of a confidence policy while "punishing" expectation-observation mismatches very selectively. Another way of looking at it is that r_e does not decrease with every mismatch, but rather only decreases in it situations for which the creature considers the expectation to have failed.

Violations and plausibility

Section 2.1.2 described the principle of grounded plausibility, by which a creature's senses determine the plausibility of an observation and thereby influence the confidence of an assumption. Later, in Section 2.1.3, we learn how basic observational plausibility can be overridden by past experience. So how does one reconcile both of these properties when dealing with expectations?

A simple example will help illustrate the solution. Imagine you are standing outside in an empty field at dusk watching a tiny glowing light approach you. Intuitively you recognize this to be a firefly. Ordinarily you might be surprised were its light to turn off because of a grounding error (you should be able to see the fly, given the assumption that it was right in front of you, but your eyes would be unable to make a measurement). But from past experience you have become accustomed to the fact that fireflies blink in and out of visibility; therefore only a drastic change in state, like seeing the firefly suddenly explode into a scorching ball of fire, would seem implausible.

The upshot of that story is that when you come to develop expectations about the behavior of objects in the world, expectations violations supersede sensory grounding as the primary arbiter of plausibility. This behavior comes about naturally as a result of the effect that violations have on salience—with each violation comes a corresponding decrease in reliability, which exerts a negative influence on expectation confidence, resulting in a lower likelihood of subsequent violations, causing the creature to become habituated to

[7]Though it approaches the true reliability as $T \to \infty$, r_e is actually only an approximation of the true reliability of the model. The difference can be quite significant for small values of T, but the approximation can be improved through statistical methods.

unusual observations.

The exception to this rule is when a measurement is not available to confirm an existing expectation (e.g., your car is around the corner and you cannot see it). In this case no comparison can be made between prediction and observation and therefore sensory grounding is used to determine whether or not it makes sense that a measurement cannot be made (Section 2.1.2).

Violations and expression

So far we have discussed expectation violations with respect to the creature's internal reasoning and behavior selection processes, but another way by which humans assess the intelligence of living organisms is through the ways they stereotypically express expectation violations[8]. Obvious examples of these types of behavior include: rubbing of the eyes for disbelief; expressing surprise by double-takes, raising of the eyebrows, widening of the eyes, and reflexively backing away from an object; head scratching as a sign of confusion, etc.

Though causally related, a violation *expression* is distinctly different from underlying expectation violation *events*. One way of thinking about this is that a brief moment of confusion is not enough to make someone act confused; because our expectations are not perfect, we are used to experiencing transient violations. Take the earlier example of the ball rolling behind a wall ... though you might be imagining the path that the ball travels while is is occluded, it is unlikely that you would *act* confused if the ball did not appear in the exact same instant you expected it to. You would probably be willing to wait some small amount of time until it is clear that the current violation events are indicative of something more than transient noise.

A reasonable way to estimate the length of this delay would be to assume that the amount of time a creature is willing to forego the expression of a violation is proportional to the confidence it has in the expectation. Then one could integrate the value of the violation event (e.g., 1 if there is a violation, 0 otherwise) over time and only trigger the expression of the violation when the integrated value has reached some threshold.

However, both the length of this "benefit of the doubt" period and the way a creature

[8]One such test is the preferential looking time procedure of [Spe85], which takes advantage of the fact that animals tend to stare longer at objects which have violated expectations. A nice example of this can be found in Marc Hauser's work on expectations in nonhuman primates [Hau98].

eventually expresses a violation are very much up to the designer, providing a convenient way to create the impression of various personality types such as skittishness, naïvité, and slow-wittedness.

2.2.3 Violation dynamics

Quite often more than one type of violation will be triggered by an unexpected observation. For example, if you were watching some normally peaceful monkeys at a local zoo when, without warning, they began going absolutely berserk, you would most likely experience surprise at the initial transition but then quickly become confused.

These terracing effects are a natural result of the way temporal variability in expectation confidences are handled, because each new violation lowers the reliability r_e. Consequently, when a reduced r_e is used in Equation 2.3 it has a depressing effect on the overall future rate of change in the confidence. As the confidence decreases, violations with higher confidence requirements such as surprise can no longer be sustained. Assuming that the state of the world does not change (i.e., the source of the violation remains unchanged) a state of simultaneous disbelief, surprise and confusion would quickly give way to one of combined surprise and confusion, which in turn would become only confusion and then eventually nothing. An observer might ascribe this effect to the character "gradually accepting the reality of the observation".

Another interesting interaction occurs between disbelief violations and the model underlying the expectation, because when a creature is in a state of disbelief it quite literally "doesn't believe what it is seeing". What this means is that in this situation the creature ignores the current observation, even if a measurement is available, and updates the model as if no measurement could be made. The immediate result is to cause the confidence to begin decaying as described above. Depending upon how the likelihood of an observation is determined, this may also cause the model to "loosen up" until the observation becomes more likely. In this situation an observer might say that the character is adjusting its expectations to account for the observed unpredictability (a form of habituation).

2.3 Review of major concepts

Before moving on it may be useful to recapitulate the major concepts presented in this chapter.

First I described the properties of *assumptions* which allow them to be used as placeholders for incomplete sensory information. Of particular importance was the discussion of how the saliency of an assumption is determined by its *confidence*; the level of confidence is subject to *temporal variability* and *temporally normalized decay*, as well as the occurrence of any *grounding errors*.

In order to address the shortcomings of assumptions I introduced the concept of *expectations*, in which assumptions are treated as predictions of the future. In order for a creature to behave realistically, it must react appropriately when its expectations are not met; these conditions are known as *expectation violations* and are related to both the confidence in the expectation and the *likelihood* of the corresponding observation.

Three such violations are confusion, surprise, and disbelief, and they affect the creature both internally and externally. Internally, violations change the *reliability* of an expectation; this in turn affects the rate of change in the expectation's confidence. The dynamics of this relationship can generate complex phenomena, such as stereotypical transitions in violation types (e.g., surprise degenerating into confusion) and the *habituation* of creatures to implausible behavior.

Finally, the external characteristic expression of violations provides a way for humans to gain insight into the reasoning processes of the creature. It is important that the creature wait the appropriate amount of time between a *violation event* and the *violation expression* because often times an isolated violation is not indicative of anything other than transient noise. It is suggested that the length of this waiting period be proportional to confidence in the associated expectation, though by adjusting the delay a designer could make the creature appear to possess a distinct personality type.

Chapter 3

Reference Implementation

The ultimate goal of my research and that of my colleagues is to learn how to build systems which exhibit intelligence at the level of real animals. The way we have chosen to pursue this goal is to build three-dimensional, animated, embodied artificial animals, paying careful attention to how emotion, motivation, movement, and learning work together to create the appearance of life. We call these creatures *synthetic characters*.

This chapter is organized into two major sections. The first section describes the behavior-based AI architecture constructed by the author; this has served as the underlying infrastructure for our research to date. The second section details how the theory of observationally-grounded expectation generation and response described in Chapter 2 was integrated into that system.

My implementation of expectations is not limited to the architecture in which it was tested. Therefore, readers who are familiar with reactive behavior-based architectures should feel free to treat the following architecture description as a reference and skip immediately to Section 3.2.

3.1 The synthetic character architecture

Synthetic characters are an excellent platform for testing theories of intelligence, in part because humans are intimately familiar with the behavior of real animals—any flaws in our theory are instantly noticeable by humans because the resulting behavior of the character appears “wrong”. Another compelling reason for choosing this domain is because we can draw upon a wealth of insight from traditional animation, a field which has already achieved great success in understanding the principles behind the illusion of life.

Ultimately, a synthetic character should be both compelling, in the sense that people can empathize with it, and understandable, in that its actions can be seen as attempts to satisfy its desires given its beliefs. In the process of learning to build this kind of character we have often found ourselves struggling with two fundamental problems. First, what kinds of properties or qualities have we, as observers, come to expect from a believable character? Second, given these expectations, what is the "right way" to go about implementing them?

This section presents an overview of the lessons we have learned from our experience to date. It begins by discussing a theory of how people go about understanding characters and then identifies some subsystems that we have found to be important for building characters that are compelling and easy to understand. Next, I will provide an overview of several approaches to these subsystems and show how, by separating out the semantic differences of each approach, we can arrive at the basic activity of each. I then describe the simple value-based framework we have developed for character construction, showing how each subsystem can be implemented with the four components of our framework.

The concepts presented here have been used to build many successful interactive experiences over the past two years. Our architecture made its debut at SIGGRAPH 98 in *Swamped!*, an interactive cartoon experience focusing on autonomous and semi-autonomous characters (Figure 3-1). In this exhibit the participants used a *sympathetic interface* [JWB+99] to influence the behavior of a chicken character with the intent of protecting the chicken's eggs from being eaten by a hungry raccoon. The raccoon character was arguably one of the most complex fully autonomous synthetic characters at the time, comprised of 84 distinct behaviors influenced by 5 separate motivational drives and 6 major emotions. In addition, the continuously changing emotional state of the raccoon was conveyed through dynamic multi-dimensional interpolation of its motion and facial expressions.

Our most ambitious project to date premiered at SIGGRAPH 99 as part of `(void *)`[1], an exhibit in which participants interact with three very different autonomous characters who are dining in a cafe (Figure 3-2). Inspired by Charlie Chaplin in the film *The Gold Rush*, participants were able to compel each of the characters to begin dancing by manipulating a novel interface consisting of small dinner rolls with forks stuck into them; the experience centered around the ways in which characters learned from and responded to the actions

[1]Pronounced "void star", the complete title is "`(void *): a cast of characters`"

of the user and of each other. This exhibit highlighted new work in the areas of emotion-based learning and motion interpolation, autonomous camera control, and dynamic music generation.

Figure 3-1: The raccoon and the chicken are two of the creatures in *Swamped!*

3.1.1 Expectations of a synthetic character

To learn how to build believable characters we can look back upon the rich history of traditional character animation. When looking at a character brought to life by a great animator we know exactly what that character is thinking and feeling at every instant and, while we may not know exactly what it is about to do, we can always call upon our perception of its desires and beliefs to hazard a guess. Even when our guess is wrong, the resulting behavior nearly always "makes sense".

Classics like *The Illusion of Life* [TJ81] explain the art of creating believable characters, which is fundamentally the art of revealing a character's inner thoughts—its beliefs and desires—through motion, sound, form, color and staging. But why do these techniques work? The American philosopher Daniel Dennett believes that they work because, in order to understand and predict the behavior of the animate objects around them, people apply what he calls the *intentional stance* [Den87]. The intentional stance, he argues, involves treating these objects as "'rational agents" whose actions are

Figure 3-2: Three characters in the diner environment of `(void *)`

those they deem most likely to further their 'desires' given their 'beliefs'" [Den98].

Desires are the key to understanding and identifying with a character. When you see the wolf look "longingly" at Little Red Riding Hood, perhaps licking his lips, you conclude that the wolf is hungry and wants to eat the heroine. How did you arrive at this conclusion? By applying the intentional stance, of course! Why else would he be acting hungry unless he *was* hungry?

Beliefs are what turn desires into actions, reflecting influences such as perceptual input ("If I see a stream, then I believe I will find water there"), emotional input ("Because I am afraid of that person, I will run away from him"), and learning ("The last time I was in this field I saw a snake, therefore I will avoid the field today"). People understand the actions of characters by inferring how their beliefs influence the ways they attempt to satisfy their desires.

How can one apply both the insights of skilled animators and knowledge of the intentional stance to build a synthetic character that people find compelling *and* understandable? From the standpoint of engineering, these expectations can be broken down into a short list of functional subsystems:

- **Motivational Drives:** For a character to appear properly motivated it must continue to work towards satisfying its desires while gracefully handling unexpected situations. For example, a creature that is starving may temporarily ignore its hunger in order to flee from an approaching predator. Once the danger has passed, however, the creature should resume searching for food. By biasing action selection towards behaviors that will satisfy the internal needs of the creature, motivational drives provide a way to achieve goal-oriented behavior.

Several researchers have addressed the problem of motivations in the context of building creatures. One example is the work of Blumberg [Blu96], who used temporally cyclic "internal variables" in the design of a virtual dog to bias action selection and facilitate external direction of synthetic actors. In another domain, Breazeal [Bre98] has developed a motivational system for regulating interactions between a robot "infant" and its human caretaker, with the goal maintaining an environment suitable for learning.

Most approaches agree on the general behavior of drives. Most importantly, they are cyclical and homeostatic—positive or negative deviations over time from the base state of "satisfaction" represent under- and over-attention, respectively, to a corresponding desire. These desires can be attended to by the successful execution of attentive behaviors like eating, or by changes in external stimuli, such as temperature fluctuations or interactions with other creatures. When unattended to, drives slowly increase over time; the effect of attentive actions is to shift the value of the drive back towards its homeostatic base state.

- **Emotions:** Emotions bias action selection in much the same way as drives. For example, a creature that is angry may be more prone to violent behavior than one who is happy. However, emotions also bias the quality of the character's motion. If the creature is sad it should walk sadly; if it is fearful it should reach for objects in a manner which conveys its fear. In this way emotion helps observers to form an empathic bond with the creature and makes its behavior appear properly motivated [TJ81].

  There are many approaches in the literature to the modeling of emotions and other affective phenomena. In so-called "appraisal" theories of emotion the individual is said to make a cognitive appraisal of their current state relative to a desired state. For example, Reilly [Rei96] proposes that fear might be modeled as proportional to "the likelihood of failing to achieve the current goal" multiplied by "the importance of not failing". Others such as LeDoux [LeD96] argue that emotions can act at a level far below the cognitive, since animals can feel emotions without consciously understanding why. Combining these approaches, Velasquez [Vel98] presents a framework that models how emotional systems interact with the perceptual, motivational, behavioral, and motor systems.

The general consensus of these models is that, instead of increasing slowly over time as do drives, emotions typically exhibit a large impulse response followed by a gradual decay back down to a base state. By altering the decay term and the gains on stimuli one can adjust the magnitude and slope of the impulse response, shaping the characteristic response of the emotion. Adjusting these parameters across the space of emotions is equivalent to shaping the "temperament" of the creature. Similarly, by altering the bias term on each emotion predisposes the creature to a particular emotional state, setting its "mood". These decay, bias, and stimulus terms represent the influences of a variety of systems[2], which in turn are affected by the current emotional state.

It is perfectly appropriate to model the influences of multiple emotions upon internal processes such as action selection, but it is difficult for human observers to visually perceive more than one emotion at a time. This is why animators tend to emphasize the most important emotion of a character, avoiding "mixed emotions". Because we are designing characters for humans to interact with, it is important for the underlying emotional model to support some notion of a "dominant" emotion. This dominant emotion can then be used to parameterize motion and expression, giving the observer insight into the internal desires and beliefs of the character.

One example of such a parameterization is the animation system of Rose, Cohen, and Bodenheimer [RCB98], in which motor commands are specified in terms of verbs ("walk", "reach-for") and adverbs ("sadly", "impatiently"). Through the use of multidimensional interpolation, this system can be used to continuously modify a character's motion in order to represent the changing state of one or more emotions (for example, making a character move as if it is mostly happy, but slightly impatient and

[2]E.g., factors include the neurobiological (e.g., hormones), motivational (intense hunger), cognitive (an impending conference deadline; the perception of a predator), and sensorimotor (posture)

somewhat tired).

- **Perception:** Fundamentally, a situated, embodied agent needs a way to "make sense" of the world in which it is situated. By this I mean two things. First, the creature needs a method of sensing the world around it; second, it must have a mechanism for evaluating the salience of incoming sensory information. The combination of a sensory stimulus and its corresponding evaluation mechanism is known as a *perceptual elicitor* or what ethologists refer to as a *releasing mechanism* [Lor73, McF73].

  Sensory information can be provided to a synthetic creature many forms, most of which fall into the three basic categories: real-world physical sensing, synthetic vision, and direct sensing. Physical devices like the temperature sensors in the motors of the Cog robot [Bro96] and the infrared sensors on the mobile robots of Mataric [Mat94] are typical of real-world sensors. Synthetic vision techniques attempt to extract salient features from a physical scene rendered from the viewpoint of the creature; examples include the ALIVE system of Maes [MDBP94, MDBP96] and the artificial fish of Tu and Terzopolous [TT94]. In direct sensing, creatures gain information by directly interrogating the world or an object within the world include; this is the approach taken by the boids of Reynolds [Rey87] and many video games.

  One of the important contributions of Blumberg, building on ideas from Lorenz [Lor73], Baerends [Bae76], and McFarland [McF73], is the notion that external perceptual influences must be reduced to a form that is compatible with internal influences such as motivations and emotions. Using a consistent internal "common currency" is essential for addressing the issue of behavioral relevance—a piece of rotting food should be as compelling to a starving creature as a delicious-looking slice of cake is to a creature that has already eaten too much. Given this representational paradigm, opportunistic behavior is simply a side effect of the relative difference in weighting between external and internal influences.

- **Action Selection:** Regardless of the particular implementation, the fundamental issues for any action selection scheme to address are those of adequacy, relevance, and coherence [Bro91a]. Adequacy ensures that the behavior selection mechanism allows the creature to achieve its goals. Relevance, as noted above, involves giving equal consideration to both the creature's internal motivations and its external sensory stimuli, in order to achieve the correct balance between goal-driven and opportunistic behavior. Coherency of action means that behaviors exhibit the right amount of persistence and do not interfere with each other or alternate rapidly without making progress towards the intended goal (i.e., behavioral aliasing).

In an effort to achieve these goals in noisy and dynamic environments, the last two decades of agent research have seen a shift away from cognitivist "Planning" approaches towards models in which behavior is characterized by the dynamics of the agent-environment interaction. In these environments, *nouvelle AI* researchers argue, collections of simple, competing behaviors that are tightly coupled with sensors and actuators can be more effective than complex planning mechanisms, while exhibiting many of the same capabilities. Examples of these approaches include the Pengi system of Agre and Chapman [AC87], the subsumption architecture of Brooks [Bro86], the spreading activation networks of Maes [Mae90], and the "Society of Mind" theories of Minsky [Min86].

In an attempt to leverage the advantages of both approaches, some hybrid systems like that of Firby [Fir87] have used a planner to make high-level behavioral decisions while using a reactive system for low-level control during behavior execution.

Inspired by ethological theories of behavior, some systems use a hierarchical organization to break complicated tasks down into specialized cross-exclusion groups [Min86] in which mutually-exclusive behaviors compete for dominance, using mutual and lateral inhibition to control arbitration [Lud76]. These include most notably the Hamsterdam system of Blumberg [Blu94] and the work of Tyrrell [Tyr93]

3.1.2 The four components of a value-based framework

In the previous section I talked about some of the important building blocks of a character that acts and emotes in a way that people find understandable and compelling. But how should one go about implementing these subsystems? In our experience we have found it useful to try a variety of approaches; this continual improvisation is made easier when the underlying framework makes it easy to implement and integrate different models.

The traditional approach to building creatures has been to focus on each of these subsystems individually. However, if we step back for a moment and consider them as a whole, two important regularities become apparent. First, there is a high degree of interdependence among subsystems—perception, emotions, and drives influence action selection, and the results of action selection in turn affect the external state of the world and the internal state of the creature. Second, the function of each can be interpreted as a quantitative mechanism. For example, the changing value of emotions and drives indicate the state of internal needs, perceptual elicitors determine the relevance of percepts, and action selection mechanisms choose the most appropriate behavior from among multiple competing ones.

What this suggests is that there is a great deal of common functionality among these subsystems. In many cases the functions performed by these subsystems can be seen as *simply different semantics applied to the same small set of underlying processes.* Consequently, instead of struggling to integrate multiple disparate models for each subsystem, it makes more sense to build them all on top of a framework that provides these shared constructs.

We have constructed this type of framework from four basic underlying components. The coherency of the framework comes from the fact that the primary internal representation is the floating-point value. In addition to being an intuitive way to think about emotions, drives, and sensory input, value-based frameworks have a number of other advantages. They are relatively easy to implement and fast at run-time, have useful parallels with reinforcement learning and neural networks, and are easily extendable because external semantics are kept separate from internal representation.

Granted, not everything is best represented numerically. However, for the purposes of getting along in the world, the processes which could potentially produce non-numeric representations (sensing and cognition, e.g.) can be seem as means to one end—action. And before any creature takes action it must first decide what action to take, which is a qualitative evaluation. Therefore, for the purposes of action selection, all semantic representations in the system are first converted to a value.

1. **Sensors:** In our system, the sensor primitive is an abstract component that operates on arbitrary input and outputs a set of objects appropriate to the sensor's functional criteria. Sensors typically use the external world or the internal state of the character as input. In addition, they may use the output of a different sensor as input; in this manner a directed, acyclic data-flow sensing network may be formed. For example, a `VisibleObjectSensor` could find all the visible objects in the world (through direct sensing, computational vision, or any arbitrary method), passing its output to a

`DogSensor` to filter out everything but dogs.

The output set of a sensor is made up of `SensorData` objects, each of which contain an object's state as described in Section 2.1.

2. **Transducers:** The transducer primitive operates on a set of input objects to produce a single floating-point output; transducers are the gateway through which sensor data enters the computational substrate. The values produced by transducers are often objective and the result of basic computations, such as the distance to the first sensed object. However, there is nothing to restrict a transducer from returning a subjective result from a complex computation—reasoning with predicate calculus about a set of input obstacles and returning the best heading in which to move, for example. Chains of sensors and transducers form the perceptual elicitors that allow the creature to react to internal and external situations.

3. **Accumulators:** The third primitive in the framework, the accumulator, is the primary unit of computation. Its inputs and gains are typically the output of transducers or other accumulators, and by constructing feedback loops it is possible to create highly connected networks which exhibit useful temporal behavior. The value V_t of an accumulator at time t for N inputs and gains is:

$$V_t = \sum_{i=0}^{N-1} \text{input}_{t,i} \cdot \text{gain}_{t,i} \tag{3.1}$$

where N is arbitrary.

4. **Groups:** The fourth primitive, the group, is used to organize accumulators into semantic groups and impose arbitrary behavior upon them. For example, a group might force the value of its accumulators to be zero except for the accumulator with the highest value. This abstraction keeps the syntax and configuration of the accumulators independent of their group semantics.

3.1.3 From components to subsystems

As an illustration I will now show one way in which each subsystem can be constructed from the components of the framework.

- **Drives:** Motivational drives can be expressed using an accumulator with a feedback loop whose gain is at least one. Attentive and aggravatory stimulus inputs are given negative and positive gains, respectively, and one additional input-gain pair represents the magnitude of the growth term. The setup in Figure 3-3 creates a drive in the style of Breazeal [Bre98].

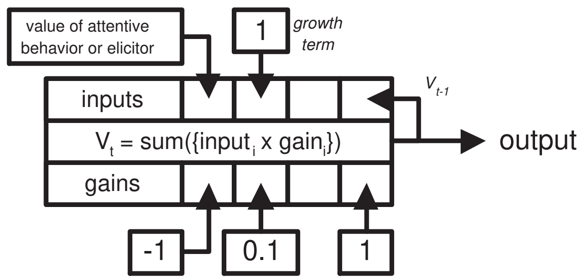

Figure 3-3: An accumulator-based motivational drive

Assuming that each stimulus$_i$ is a positive-valued stimulus working to satiate the drive, this configuration increases in value over time from a homeostatic base state of zero, according to (3.2).

$$V_t = V_{t-1} + \text{growth}_t - \sum_i \text{stimulus}_{t,i} \tag{3.2}$$

- **Emotions:** Emotions can be implemented with a configuration similar to that used for drives where, instead of acting as a growth term, the input-gain pair biases the homeostatic base state. By limiting the gain on the feedback loop to the range $(0, 1)$ we can effect a gradual decay over time in the value of the emotion. This configuration, show in Figure 3-4, varies in time according to (3.3).

  Often it is useful to organize emotions into cross-exclusion groups for the purposes of identifying the dominant emotion. By adjusting the inhibition between the competing emotions one can tailor the personality of the creature—making a fearful creature less prone to happiness, for example.

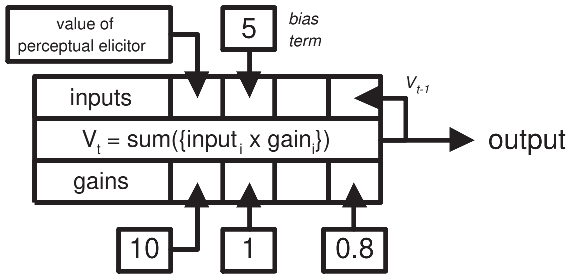

Figure 3-4: An accumulator configured as an emotion

$$V_t = (V_{t-1} \cdot \text{decay}_t) + \text{bias}_t + \sum_i \text{stimulus}_{t,i} \tag{3.3}$$

- **Action Selection:** A behavior is simply an accumulator that is semantically associated with a particular behavioral routine that it executes while "active"; typically this involves sending a message (e.g., "walk") to an underlying motor system. Their inputs are the outputs of emotions, drives, and perceptual elicitors; whether a behavior is considered active or not is determined by the semantics of its associated group. For example, autonomic behaviors like breathing and blinking might be contained in a group whose policy is to activate any behavior with a value above a certain threshold.

  To achieve ethologically-inspired action selection policies, mutually exclusive behaviors can be organized into groups with cross-exclusion and mutual inhibition semantics and forced to "compete" on the basis of their output values. Hierarchical action selection in the style of Blumberg and Tyrrell is easily implementable by associating each behavior with a reference to another group.

  This method of implementing action selection has the advantage of making behavior design independent of action selection policy, allowing the designer to use the same behavior in many different contexts. For example, under normal circumstances a character might execute a swallowing behavior at regular intervals; this same behavior, however, might be a sub-behavior with an explicit order in the context of eating a meal. In our framework the same behavior can be used in both situations without requiring the designer to implement or have *a priori* knowledge of policy-specific details (e.g., connections to parent behaviors, execution order, etc.). This flexibility facilitates creating libraries of generic behavioral routines from which a variety of characters can be constructed.

3.2 Integration of expectation generation and response

In many respects synthetic characters are a perfect vehicle to test a system for expectation generation and response. In entertainment, for example, the kinds of characters we would like to interact with are of cartoonish exaggerations of real animals, and much of classic cartoon humor—Daffy Duck's double-take of disbelief after running off a cliff; Wile. E. Coyote's surprise in discovering that his cigar is actually a stick of dynamite; Elmer Fudd's confusion when Bugs Bunny taps him on the shoulder and then disappears—are based upon expectation violations. In general, any kind of character you might want to tease or trick needs object persistence and expectations.

To integrate expectations into the synthetic character architecture described above I have created a new type of sensor which, when inserted into the sensory network, manages the creation, persistent updating, and eventual elimination of expectations about sensory information received at some point in time up to the present. In the implementation, and in the rest of this section, this sensor is called the `ObjectPersistence` module (OPM), and the data structure that represents and embodies an expectation is referred to as a *persistent object*.

The OPM works by taking in measurements of the state of objects over time and providing as output predictions of what the state of previously observed objects will be in the very near future[3]. In the following sections I will describe this process by which this happens in more detail.

3.2.1 The `ObjectPersistence` module interface

The designer of the character specifies the creature's behavior by linking together the components discussed in Section 3.1.2 and modifying their parameters in order to build more complex higher-level subsystems. Though these connections and parameters can be adjusted

[3]For any such object for which a prediction can be made with a non-zero confidence. The special case is when a `reject-1` grounding error occurs, because the creature should not react to an expectation that is disproved by its sensory grounding.

from within Java (the programming language in which the architecture is implemented), the usual manner in which characters are specified is via an ASCII text file whose format describes the relationship between value containers, or *nodes*, and the values themselves, called *fields*. This format is loosely based upon that of the Virtual Reality Modeling Language (VRML)[4].

Within this file the format of the OPM is:

```
ObjectPersistence {
   VrmlSFNode     input
   VrmlSFFloat    minimumAllowedLikelihood
   VrmlSFFloat    lowConfidence
   VrmlSFFloat    mediumConfidence
   VrmlSFFloat    highConfidence
   VrmlSFFloat    veryHighConfidence
   VrmlSFFloat    maxConfidenceDecayRate
   VrmlSFFloat    minConfidenceDecayRate
   VrmlSFFloat    maxConfidenceGrowthRate
   VrmlSFFloat    minConfidenceGrowthRate
   VrmlSFNode     featureGrounding
   VrmlMFString   salientFeatures
   VrmlMFNode     featureUpdateMethods
}
```

where the `VrmlSF` prefix denotes a single-valued field and the `VrmlMF` prefix indicates a multiple-valued field, which is the equivalent of an array. Thus, a `VrmlSFFloat` is a field whose value is a single floating-point number, and a `VrmlMFNode` field contains an array of nodes. The OPM is itself a VrmlNode.

Each of these fields represents a parameter by which the designer may control the way in which expectations are generated and the conditions under which violations will occur. The nature of each parameter is as follows:

`input`: The input to this the OPM, which must be a Sensor

[4] International Standard ISO/IEC 14772-1:1997

`minimumAllowedLikelihood`: The minimum allowed likelihood for an observation to be considered "normal" given the prediction. If the likelihood is less than this value, then the observation does not match the prediction (i.e., "it's too weird").

`lowConfidence`: The level of confidence (in the model of a feature) that is considered "low". The confidence needs to be at least this high in order for the creature to be confused.

Restriction: $0 <$ `lowConfidence` $\leq$ `mediumConfidence`

`mediumConfidence`: The level of confidence (in the model of a feature) that is considered "medium". The confidence needs to be at least this high in order for the creature to be surprised when there IS an observation available.

Restriction: `lowConfidence` $\leq$ `mediumConfidence` $\leq$ `highConfidence`

`highConfidence`: The level of confidence (in the model of a feature) that is considered "high". The confidence needs to be at least this high in order for the creature to be surprised when there is NOT an observation available.

Restriction: `mediumConfidence` $\leq$ `highConfidence` $\leq$ `veryHighConfidence`

`veryHighConfidence`: The level of confidence (in the model of a feature) that is considered "*very* high". The confidence needs to be at least this high in order for the creature to be ever be in a state of disbelief.

Restriction: `highConfidence` $\leq$ `veryHighConfidence` ≤ 1

`minConfidenceDecayRate`: This is the minimum rate, in units of second^{-1}, which the creature's confidence in a feature will decay.

Restriction: $0 <$ `minConfidenceDecayRate` $<$ `maxConfidenceDecayRate`

`maxConfidenceDecayRate`: This is the maximum rate, in units of second^{-1}, which the creature's confidence in a feature will decay.

Restriction: $0 <$ `minConfidenceDecayRate` $<$ `maxConfidenceDecayRate`

`minConfidenceGrowthRate`: This is the minimum rate, in units of second^{-1}, which the creature's confidence in a feature will grow.

Restriction: $0 <$ `minConfidenceGrowthRate` $<$ `maxConfidenceGrowthRate`

`maxConfidenceGrowthRate`: This is the maximum rate, in units of second^{-1}, which the creature's confidence in a feature will grow.

Restriction: $0 <$ `minConfidenceGrowthRate` $<$ `maxConfidenceGrowthRate`

`salientFeatures`: The names of all the features in the incoming sensory data for which expectations should be generated. All other features are ignored and will not be present in the output.

`featureGrounding`: The PerceptualGrounding object which implements the sensory grounding function described on page 20; a common grounding, for example, the creature's visual sensor.

`featureUpdateMethods`: The mechanism(s) used to update the value of the salient features with measurements from new observations. If these are left unspecified, the default behavior is to keep the value of each feature the same in the absence of a new measurement, and to update them from a new measurement by directly copying the value of the measurement.

3.2.2 High-level overview of the algorithm

Sensory data enters the OPM as an array of state measurements $\{S(O_0)\,, \ldots, S(O_{n-1})\,\}$, where each state $S(O_i)$ corresponds to one of the n objects perceived at the current time $t = t_{now}$. This set of observations is then compared the set of pre-existing persistent objects in order to determine the appropriate way of handling the new state measurement. The three possible cases and their corresponding actions are as follows:

1. **Case:** The measurement does *not* correspond to any existing persistent object.

   Action: Create a new persistent object to corresponding to the measurement (Section 3.2.3).

2. **Case:** There exists a persistent object that corresponds to the measurement $S(O_i)$.

   Action: Update the persistent object, taking into account the new measurement (Section 3.2.4).

3. **Case:** There exists a persistent object for which no new measurement is available.

   Action: Update the persistent object, taking into account the fact that a measurement is unavailable (Section 3.2.4).

After all of the pre-existing persistent objects have been updated and, if appropriate, new persistent objects have been created, the output of the OPM is an array of persistent objects $\{P(O_l), \ldots, P(O_m)\}$, where $(m-l)+1$ is the current number of persistent objects in the OPM and $0 \leq l \leq m \leq max(\forall t \leq t_{now} : \; n_t)$. Each of these persistent objects represent an expectation of what the state of object O_i will be in the very near future. To the rest of the behavior system, these persistent objects appear no different from any other type of sensory data because their structure and method of manipulation is identical to that of normal sensory information.

It should be noted the problem of determining correspondence between a new measurement and an existing persistent object is handled by direct pointer comparison. The persistent object data structure retains as a reference a pointer to the real-world object that it represents. If an object in the incoming sensory data matches the reference pointer in a pre-existing persistent object, then correspondence has been achieved. This is not as cognitively plausible as the other mechanism proposed earlier (page 23), but perfect correspondence does not seem to sacrifice any realism. My guess is that this is because correspondence is one of the sensory-cognitive abilities (like object persistence and expectations) that are so fundamental as to be unnoticeable when functioning properly[5].

3.2.3 Creating new expectations

Creating a new expectation involves creating a new persistent object. The expectation's initial state is created by recursively traversing the measurement and copying the value of its features. However, the addition of each feature for which predictions are made brings with it a linear decrease in system performance. Therefore only those features that the designer has declared as salient (specified by name in the `salientFeatures` field of the OPM) or that are necessary for mimicking the structure of the measurement are made part of the persistent object's state. A pseudo-code description of this recursive process is shown in Figure 3-5.

In addition to initializing feature values, this creation phase also assigns each salient feature the mechanisms by which both the prediction of the feature's value and the confidence in that prediction are updated. The feature update method assigned is determined by the

[5] Of course, one could apply the same argument I made in Section 1.1.3 against me by saying that a synthetic creature will not behave realistically until it makes the same correspondence errors as real animals.

value of the `featureUpdateMethods` field of the OPM, as described earlier; the confidence policy assigned is an `EVBasedConfidence`, which will be discussed in the next section. This information is stored within each persistent object and is invisible to inspection outside the OPM.

```
FUNCTION RECURSE_CREATE(observedState, State newPersistentObject) {
  FOREACH feature IN observedState {
    IF (feature IS_A State) {
      // This feature must be copied, both to mimic the
      // structure of the observation, and to ensure that
      // any salient features within it are copied.
      State recursiveCopyOfFeature = NEW State;
      RECURSE_CREATE(feature, recursiveCopyOfFeature);
      ADD_FEATURE(newPersistentObject, recursiveCopyOfFeature);
    }
    ELSE IF IS_SALIENT(feature) {
      // Copy this feature because it is salient
      State copyOfFeature = CLONE(feature);
      ADD_FEATURE(newPersistentObject, copyOfFeature);

      // Assign the update mechanisms for the feature's
      // value and confidence
      ASSIGN_FEATURE_CONFIDENCE(newPersistentObject, copyOfFeature);
      ASSIGN_FEATURE_UPDATER(newPersistentObject, copyOfFeature);
    }
    ELSE {
      // Don't copy this feature, because it is not a
      // salient feature or state
    }
  }
}
```

Figure 3-5: Pseudo-code for the recursive method used in creating new persistent objects

3.2.4 Updating existing expectations

Each time the OPM is queried for new data by the other modules in the creature's sensory chain, all pre-existing persistent objects $P(O_i)$ are updated to reflect either a new state measurement $S(O_i)$, if available, or the lack of a such a measurement. This process is as follows (steps 1d through 1g are discussed in more detail later in this section):

1. Beginning with the root, both $P(O_i)$ and $S(O_i)$ are recursively traversed in parallel. At each level in the persistent object hierarchy the following actions[6] are taken in

[6]Though it has been left out for the sake of clarity, all functions also include time as a variable

sequence:

(a) For each salient feature f an attempt is made to get the value of the corresponding feature f' in the state measurement. I will indicate that f' is not available by saying that $A(f') =$ false; otherwise $A(f') =$ true.

(b) The persistent object is queried for the feature value and feature confidence mechanisms associated with f. These will be referred to as U_f and C_f, respectively.

(c) The sensory grounding (specified in the `featureGrounding` field of the OPM) is queried to determine if it should be possible to obtain the measurement f' given the expectations about the state $S(O_i)$ embodied in $P(O_i)$; call this result $V(P(O_i))$. For example, if the grounding is visual, $V(P(O_i)) =$ true if and only if $P(O_i)$ represents an object that should be visible to the creature at the current time.

(d) U_f is queried to to determine the likelihood $L_{f'}$ of the measurement f' given the expected value f and the grounding information $V(P(O_i))$.

(e) U_f is used to update the expectation f such that $f_{t+\delta} = U_f(f_t, f'_t)$. Because the current version of our architecture updates in discrete intervals, δ is a small positive value around on the order of 30 milliseconds.

(f) The OPM determines whether or not any expectation violations EV_f have occurred for this feature, taking into account $V(P(O_i))$, $L_{f'_t}$, $A(f')$, f', and the current confidence in the predicted value of the feature, c_f.

(g) C_f is used to update c_f such that $c_{f_{t+\delta}} = C_f(f'_t)$. In the special case of the expectation-based confidence metric described later, the expectation violations are included as additional arguments to C_f.

2. If $\{\forall f : f \in P(O_i) \mid c_f = 0\}$, meaning that no feature with a non-zero confidence remains, then $P(O_i)$ is permanently deleted from the OPM.

3. $\forall f : f \in P(O_i)$ any violations EV_f are registered with the OPM so that they may be detected within the creature's behavior system.

After each persistent object has been updated, checks are made to determine if it should be added to the output of the OPM:

1. If a state measurement $S(O_i)$ was available, then add $P(O_i)$ to the output[7].

2. If the updating of $P(O_i)$ triggered a disbelief violation then $P(O_i)$ is added to the output[8] since the creature should literally "not believe what it sees" and trust its expectation rather than its senses.

3. If a state measurement $S(O_i)$ was *not* available, then $P(O_i)$ is only added to the output if $V(P(O_i))$ is false. Think of this using the example of the ball rolling behind the wall: since the ball cannot be seen behind the wall, the creature should only react to the expectation of the ball's position while the ball is not supposed to be visible (i.e., while it is hidden by the wall). To the observer, this looks like the creature is "guessing" the ball's location. Unless the creature were in disbelief (as already described), if the ball were to fail to roll out from the other end of the wall as expected (in which case $V(P(O_i))$ would be true but no measurement would be available, triggering a `reject-1` grounding error), the creature should *not* react to its expectation. This would result in incorrect behavior, appearing to the observer as if the creature was too stupid to realize that the ball had not appeared.

As a final step, any expectation violations which occurred in the past but are no longer valid are removed from the OPM and the creature's behavior system.

The feature updating mechanism

One of the fundamental requirements of the expectation model is the ability to update an expectation given new observational data. This is a very difficult and open-ended problem that I view as being very feature- and task-specific. Therefore in my implementation I have chosen to give the designer the ability to specify the appropriate mechanism on a per-feature basis. Each updating mechanism implements a particular function interface which allows the OPM to pass as input the value of the feature in the expectation, f_t, and the current

[7]This is incorrect behavior in the event of a `reject-2` grounding error (i.e., the creature can observe something which should be impossible to observe given its knowledge of the world or a corresponding expectation) but, as noted on page 19, my theory does properly handle this situation.

[8]This implementation is somewhat flawed in that if any feature $f \in P(O_i)$ causes a disbelief violation then the creature effectively adopts a position of disbelief towards $P(O_i)$ in its entirety. This does not make sense in many cases; e.g., I could be in disbelief regarding the clothes you are wearing, yet still believe what my senses tell me about your position in space. This problem has not yet been noticeable, however, because none of the creatures we have built to date are complex enough to care or convey disbelief about details as subtle as a style of clothing.

measurement of that feature, f'_t. The output of the function is the updated expectation of what the feature's value will be at some point in the near future, $f_{t+\delta}$.

My implementation currently includes two updating mechanisms. The first is the `DefaultUpdateMethod`, which simply sets the value in the expectation to be the same as the measured value if a measurement is available. This is useful when it is important that a feature be declared salient so that it becomes part of the expectation state (and hence is available to the rest of the sensory hierarchy), but the designer does not care about detecting expectation violations for that feature.

The second mechanism I have implemented is called the `KalmanFilterUpdateMethod` and is based on the discrete Kalman filter approach to optimal estimation[Gel74, WB95]. Given noisy and discrete samples of one or more of the variables in the state of a continuously-varying linear function, the Kalman filter attempts to approximate the true value of the function. I chose this method, not because I believe animals necessarily use statistical methods of estimation, but rather because the Kalman filter has the following desirable qualities:

1. It can estimate the process state while lacking observations of all state variables (e.g., estimating position, velocity, and acceleration through observations of position alone)
2. It can be used to "predict ahead" an arbitrary number of time steps
3. Estimation with gracefully degrading accuracy is possible in a temporary absence of observations
4. It is capable of providing an estimate of its error in the approximation of each state variable

These qualities are desirable because they mimic qualitatively some of the abilities which humans and other animals seem to possess.

Determining the likelihood of a measurement

As was discussed in Section 2.2.1, an expectation violation can occur in one of two situations, the first being when an observation causes a grounding error and the second when the difference between the predicted value of the feature and the measured value is highly unlikely. Because the feature updating mechanism is responsible for providing the prediction of a feature's future value, it is also in the best position to answer the question "how surprising is it that I observed f' given that I had expected f?"

In the case of the `DefaultUpdateMethod`, it is assumed that the designer is not concerned with (or seeks to actively discourage) expectation violations for the associated feature so, when asked for the likelihood of a measurement, a value of 1 is always returned. The `KalmanFilterUpdateMethod` uses a more sophisticated likelihood measurement. Because part of the Kalman filter's internal state is a covariance matrix for the state estimate, we can compute the likelihood L of a given observation under the corresponding gaussian:

$$L = P(O \mid E, M) \tag{3.4}$$

$$= \frac{1}{\sqrt{(2\pi)^{\,d}\,|M|}}\, e^{-\frac{1}{2}\,(O-E)^T M^{-1}\,(O-E)} \tag{3.5}$$

where O is the state measurement vector, E is a vector containing the predicted (expected) value of the state, M is the Kalman filter error covariance matrix, and d is the dimensionality of both E and O.

It was my intent to allow the designer to specify a threshold to which likelihoods would be compared; observations with likelihoods below the threshold would be considered "unusual" and might possibly contribute to a violation. Unfortunately, in many cases the gaussian was very flat, making it difficult to distinguish between "unusual" and "acceptable" observations. To compensate for this I normalized the likelihood by its maximum possible value, which occurs when $O = E$, causing the fractional term in (3.5) to drop out[9]. This tends to overly exaggerate small differences in the true likelihood, but since the OPM is only concerned with the tail ends of the probability distribution this has proven to be a reliable metric.

A better solution would have been to use a threshold based on how many standard deviations away the measurement was from the prediction.

[9]The absolute value of the expression in the exponential term is known as the Mahalanobis Distance. Its advantages over Euclidean distance include automatically accounting for scaling of the coordinate axis and correcting for correlation between components. Though it can be unreliable in the case where components are uncorrelated, this is not a problem in my implementation because the components of a feature's state are always at least somewhat correlated (i.e., under normal circumstances the x, y, and z coordinates in a position vector cannot vary truly independently).

Detecting expectation violations

Given a measurement f', its availability $A(f')$ and likelihood $L_{f'}$, the confidence c_f in the expectation and its sensory grounding status $V(P(O))$, the OPM determines whether or not the measurement has caused one or more violations in the following manner:

1. If $A(f')$ = true and $L_{f'} <$ `minimumAllowedLikelihood` then
   (a) if $c_f \geq$ `lowConfidence` then a confusion violation has occurred
   (b) if $c_f >$ `mediumConfidence` then a surprise violation has occurred
   (c) if $c_f >$ `veryHighConfidence` then a disbelief violation has occurred
2. If $A(f')$ = false and $V(P(O))$ = true then
   (a) if $c_f \geq$ `lowConfidence` then a confusion violation has occurred
   (b) if $c_f >$ `highConfidence` then a surprise violation has occurred
   (c) if $c_f >$ `veryHighConfidence` then a disbelief violation has occurred

otherwise no violation has occurred. Note that more than one type of violation can be generated simultaneously by the same feature.

Updating the expectation confidences

As stated in the previous section, the confidence mechanisms created by the OPM for each new feature are instances of `EVBasedConfidence`. This confidence mechanism implements the reliability- and expectation-based confidence policy of Section 2.2.2 by using a collapsed version of Equations 2.3-2.4:

$$c_t = c_{t-1} + \phi * dt$$

where

$$\phi = max(\ r_e * \texttt{maxGrowthRate},\ \ \texttt{minGrowthRate}\,)$$

when $A(f')$ = true and the measurement did not cause a violation;

$$\phi = -1 * max(\ (1 - r_e) * \texttt{maxDecayRate},\ \ \texttt{minDecayRate}\,)$$

in all other cases.

3.2.5 Expression of expectation violations

Here I will discuss some implementation-specific details on how information provided by the OPM might be used by the designer of a creature.

Violation sources

Every feature in every object has the potential to be the source of one or more expectation violations. For this reason it is useful for a creature designer to have access to this information for the purposes of building violation responses tailored to specific situations or types of situations. For example, the designer might want the creature to do a double-take in most surprise situations, but respond by rubbing its eyes “whenever the color of Peter’s shirt is so unexpected that it causes disbelief.”

To account for this, my implementation keeps track of the situation that caused each violation and makes it available to the designer through a special sensor called `ExpectationViolationSources`. This sensor allows the designer to build history queries comprised of logical conjunctions of criteria such as the types of violations that occurred, the names of the features associated with those violations, and information specific to the objects containing those features.

The output of these queries are the expectations corresponding to the objects containing the violating features. This allows the designer to use violation information throughout the architecture and orchestrate complex situated responses such as “get the closest object that is red in color and whose position caused surprise; look at it and back away slowly.”

Salience

Because expectations must exhibit the property of transparency they can be evaluated using the same mechanisms a creature would use to evaluate raw sensory data. However, as Section 2.2.2 made clear, expectation confidence is the fundamental variable through which expectations can have an impact on a creature’s behavior. For example, if there was food in front of me and at the same time I had an expectation that there was food behind me, I could base my decision in part on the relative merits of the two food sources. All other things being equal, however, the decision would ultimately rest on on how confident I was that the food would actually be there if I turned around.

To allow the designer access to this information, the OPM includes a feature's confidence (in the form of a "hidden" feature) along with it in the expectation state hierarchy. The designer can then extract this confidence and use it to weight the saliency of the corresponding feature.

Violation events versus violation response

In addition to talking about saliency, Section 2.2.2 also discussed the necessity of having a "benefit of the doubt" period between the initial onset of an expectation violation and the characteristic expression of that violation. This is easily achieved in our architecture by integrating violation events using the same principles from upon which motivational drives are based, where the input to the integrator is the cardinality of the output of the `ExpectationViolationSources` module. The parameters of each "drive to express confusion, surprise, disbelief" can then be tweaked to create a creature that responds as desired in a given situation.

3.3 Evaluation

The implementation was tested in a three-dimensional world consisting of a flat square plane upon which rested a small red cube and three walls (Figure 3-6). The bipedal test creature directly sensed this world at a rate of 30 ± 10hz through a visual filter which detected objects within a $60°$ bi-directional field of view. Occluded objects were not visible to the creature and objects were not allowed to interpenetrate. The creature's only goal was to to approach the cube and stop.

The user had absolute control over the position of the cube and was able to move it in either a continuous ("sliding") or discrete ("transporting") fashion. By moving the cube around the world it was feasible to test the behavior of the creature in various situations both with and without the benefit of the `ObjectPersistence` module.

Without the OPM the behavior of the character in response to perceptual discontinuities was quite unrealistic, exhibiting the problems discussed in Section 1.1. When the cube moved behind a wall and became occluded, for example, the creature immediately stopped pursuing it (as if the cube had ceased to exist). When the cube reappeared from behind the wall the character once again took up pursuit, but displayed no outward indication

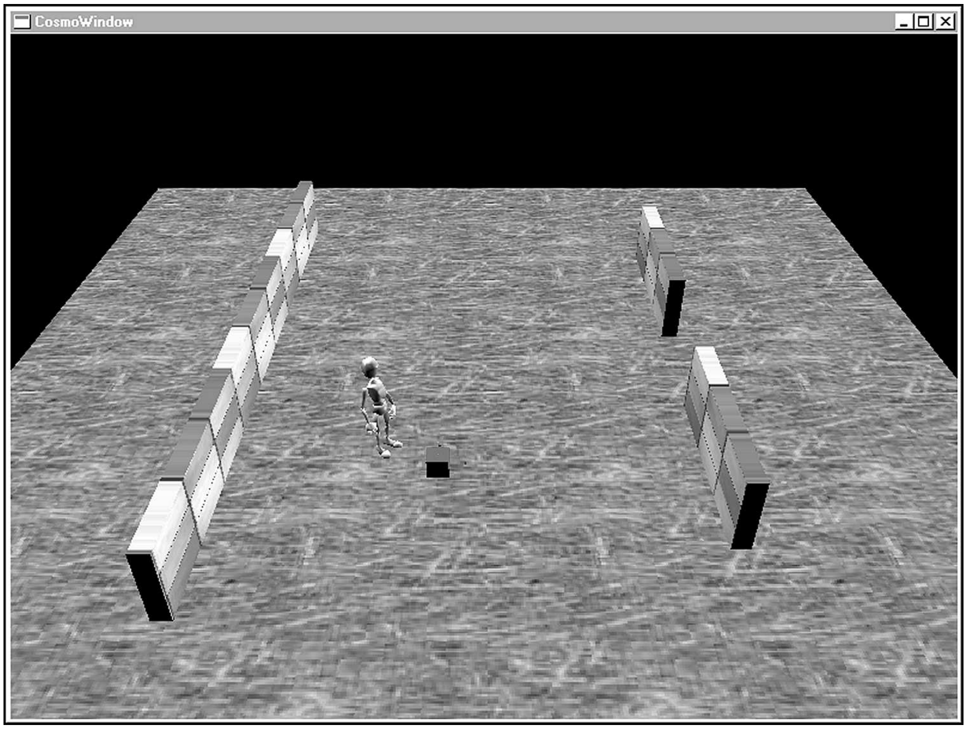

Figure 3-6: The test environment

that anything unusual had occurred. In addition, no reaction was made by the character in response to non-occluded discontinuous motion.

After the OPM was enabled, the level of realism in the character's behavior increased markedly (see Figure 3-7). For example, the character continued to pursue the cube even after it moved behind a wall. By generating an expectation of the cube's position—extrapolated from the cube's motion prior to occlusion—the creature was able to determine where it "thought" the cube would re-appear and then take the shortest path to the predicted appearance point (e.g., the character would immediately head to the other end of the wall).

These expectations also made it possible to "trick" the creature by stopping the cube after moving it behind a wall. Because the creature was unable to see the cube stop, it assumed that the cube would appear on the other side of the wall and was therefore experienced one or more violations when it did not; these violations exhibited the various types of dynamic behavior described in Section 2.2.3, such as habituation and terracing.

It was also possible to cause expectation violations without the use of occlusion. Violations were generated whenever the cube's motion was not sufficiently continuous, such as when the user made the cube instantly jump to a different location, or when the cube's acceleration was discontinuous.

The creature expressed non-transient violations through stereotypical animated behaviors. For example, disbelief was indicated by the creature rubbing its eyes ("I don't believe what I am seeing!") and then peering forward intently. Surprise was expressed through a small "startled jump" and confusion was communicated by the creature scratching his head and looking around.

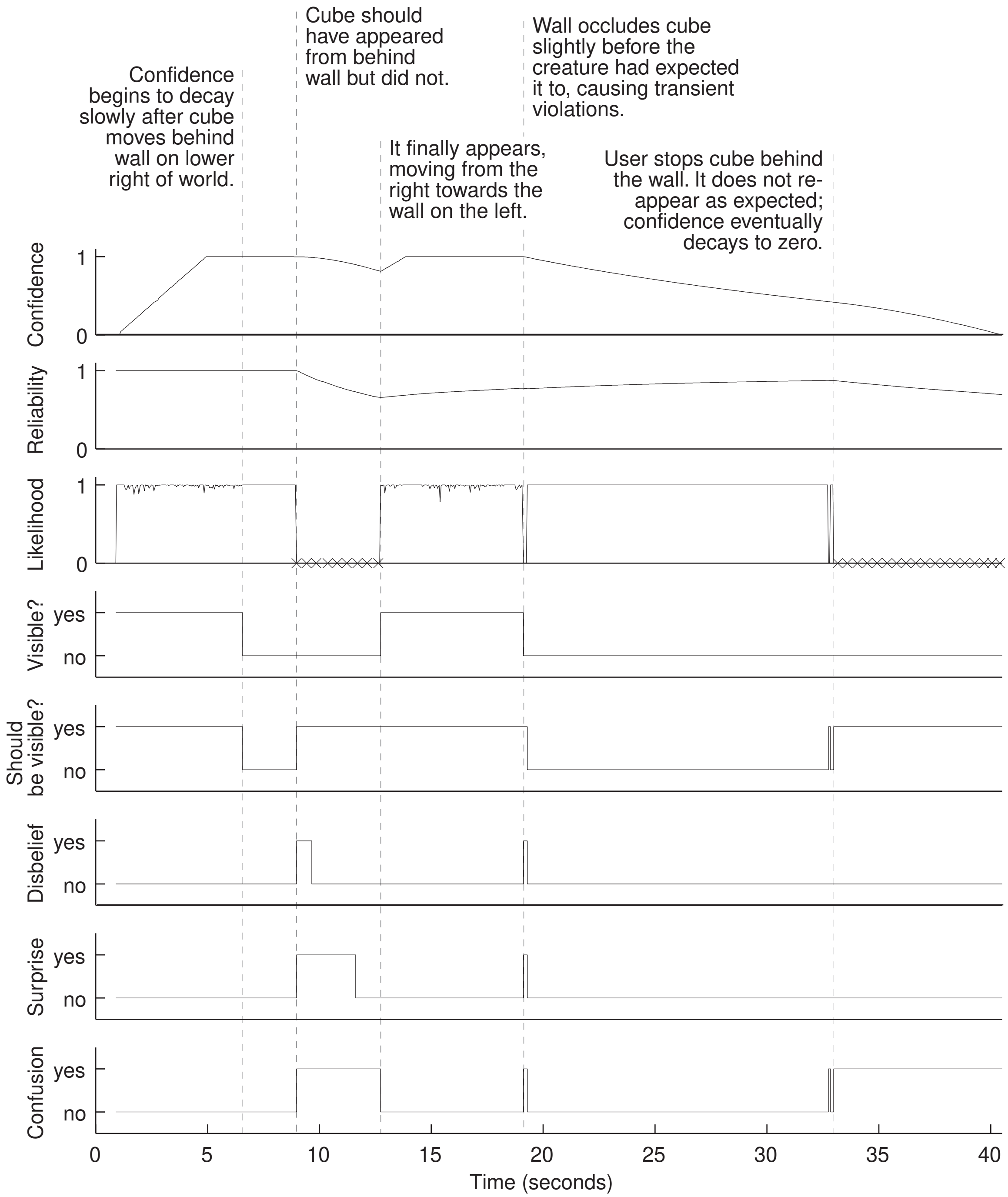

Figure 3-7: An annotated example of one test session in which the OPM was enabled. An 'X' indicates where the likelihood was undefined (discussed in Section 4.1.2)

Chapter 4

Future Work

In many ways the work that I have presented here is incomplete. Therefore in this section I would like to touch briefly upon some of the ways in which it might be improved.

4.1 Immediate issues

In addition to problems previously discussed, there are several areas in which my theory and the current implementation of it stands to be improved.

4.1.1 Problems with the theory

1. Rather than using arbitrary thresholds, it might make more sense to specify the conditions for confusion, surprise, and disbelief in terms of the ratio of confidence in the expectation to the likelihood of the measurement. For example, under normal circumstances it is highly unlikely that any given object will explode; therefore when an object does explode it is almost always surprising, regardless of confidence. In contrast, when one is in the middle of a battlefield the likelihood of an object exploding increases drastically, and therefore a higher confidence is necessary before an exploding object will be cause for surprise.

2. The likelihood of a measurement only affects confidence to the extent that unlikely measurements may cause violations and lower the reliability of the expectation. It might also make sense to weight the scaling factors in Equation 2.3 by some value proportional to the likelihood, in such a way as to make the confidence grow more

quickly when observations closely match predictions, and vice-versa.

3. Much of the behind-the-scenes knowledge in the feature updating mechanisms remains unused. Specifically, in the case of `reject-1` grounding errors it would useful to provide the creature with information about where the object might be found even though it cannot be seen. In many cases this information may already be present in the feature updating mechanisms since they are already capable of assessing the likelihood of a measurement given an expectation.

4. Under the current theory there is no consistent mechanism for handling `reject-2` grounding errors. Because of this, a creature cannot be surprised when something for which it does not already have an expectation appears "out of thin air", and certainly cannot handle the "pink elephant" case of page 19. This is one of the more glaring inadequacies of the existing theory—my creatures would not be surprised were random objects to inexplicably fall from the sky.

5. The theory cannot account for certain transitions between violation types. If your chair disappeared out from under you, for example, you might initially be surprised before entering a state of disbelief. In this case the failure of the theory hinges upon the assumption that the two violations come from the same source; however, this might not be the case—perhaps your surprise stems from the collision of your posterior with the floor while your disbelief is related to the disappearance of the chair.

6. The theory does not handle any of the problems associated with inductive reasoning (Section 4.2.1).

4.1.2 Problems with the implementation

1. The feature update methods currently implemented do not use negative knowledge to improve the expectation. To use (yet again) the example of a ball rolling behind a wall, if the ball does not roll out the other side as expected (a `reject-1` grounding error), the creature should be able to reason that "since the ball is not here where I predicted it to be, it must be somewhere in between here and the place I last observed it."

2. The feature update methods do not handle discrete-valued features very robustly. The usefulness of my implementation would benefit greatly from a mechanism that can set the value of an expectation's feature to $\mathrm{argmax_i\, P(f' \mid U_f)}$ given the i discrete feature values previously observed.

3. The algorithm used to detect expectation violations should use standard deviation rather than likelihood to classify the discrepancy between predictions and measurements.

4. Likelihood of a measurement is not able to be determined in the case of a `reject-1` grounding error. This has not been a problem for the current implementation because it does not use likelihood for detecting violations when $A(f') =$ false and $V(P(O)) =$ true. However, a change in the violation detection algorithm might necessitate this information.

5. The designer must explicitly tell the creature the features for which is should build expectations (by specifying them in the `salientFeatures` field). This could be improved by incorporating a mechanism by which a creature could learn which features are salient through observation; one example of such a mechanism can be found in [WB99].

4.2 Areas for further exploration

There are two areas related to this work which I consider to be particularly strong candidates for further research.

4.2.1 Expectations and inductive reasoning

One of the original goals of this work was to give creatures the ability to generate higher-order expectations about the behavior of objects in the world. For example, if I built a synthetic cat and set it to the task of chasing a real mouse, I would like it to begin by being able to generate the kind of first-order expectations described in the previous chapters. Building upon that, the cat should then begin to construct chains of first-order expectations to arrive at second-order expectations; this, for example, would give it the ability to predict more complex activity such as the periodic nature of circular movement or a zig-zag pattern.

The hope is that this bottom-up approach to the construction of higher-order expectations would eventually create a huge knowledgebase of experience from which to exert top-down influence upon the expectation generation process, allowing the cat to predict increasingly complex behavioral sequences given a relatively small number of observational clues. Many non-trivial issues obstruct the path to this goal:

- These additions would require a general mechanism for pattern recognition. But when does a pattern start and when does it end; by what features is it characterized, and how does one know when a new pattern is being observed? These are incredibly difficult problems.

- There must be a way to reconcile the *a priori* knowledge embodied in the feature updating mechanisms with the newly acquired information, and that knowledge must be represented in such a way as to facilitate both bottom-up generation and top-down influence.

- The creature must be able to generate expectations for an arbitrary amount of time δ in the future, and therefore needs a heuristic by which to pick the right value of δ in any given situation (knowing when to zig if expecting a zag, so to speak).

- The importance of context grows in parallel with the influence of experience in guiding the generation of higher-order expectations. In order to make the right predictions, the creature must begin taking into account how both its own actions and the state of the world will affect the behavior of the object for which expectations are being generated. This would add a tremendous amount of sophistication to our creatures' reasoning ability, allowing them to perform the kind of speculative internal simulations that is often associated with human-level intelligence. Of course, intelligence has its price—the addition of such causality would force the implementor to deal with the infamous "frame problem".

This top-down influence upon expectations is vital to intelligent behavior. For example, under the expectation model I have proposed, a creature can become conditioned to something unexpected, like a desk moving of its own accord. However, this conditioning never becomes a top-down prior for when a new desk is seen; the creature would be equally surprised if the new desk moved.

This ability to draw general conclusions from specific examples, known as *inductive reasoning*, is one of the great unsolved problems in any comprehensive theory of mind. Drescher [Dre91] addresses some of these issues at the developmental level, though it is not clear that his or any other proposed system is capable of scaling with the speed and grace seen in humans and other animals. I do not claim that I have any definitive solution to this problem, but I believe that it would be interesting and beneficial to explore the level of inductive reasoning necessary to create an artificial animal whose behavior is sufficiently convincing to a human observer.

4.2.2 Use in reinforcement learning

It is possible that expectations and expectation violations might improve reinforcement learning in the following areas:

1. **Hidden state:** One of the major problems in reinforcement learning is that an agent needs to have complete knowledge of the context order to learn which elements of that context are most relevant to the consequent utility of an action. Often, however, knowledge of the context is limited by the restrictions of perception (e.g., you cannot see what is happening behind you); compensating for incomplete perception is known as the problem of *hidden state*. The use of expectations could help in this case by "filling in" the hidden portions of context with assumptions about what their values might be[1].

2. **Exploration versus exploitation:** Another problem in reinforcement learning is knowing when to use an action that was successful in the past for a particularly context (exploitation) versus when to try different actions (exploration) in the hope that they will be more successful. Expectation violations might help with this in two ways: first, as an indicator that the creature's understanding of the world is less than complete, thereby motivating further exploration of the state space; second, because

[1] A second aspect of the hidden state problem is that there may be important features in the context that your sensors are simply incapable of detecting (e.g., human eyes cannot detect radiation in the infra-red portion of the spectrum). Expectations would be of no help in this case.

each violation is associated a particular feature, the creature knows what it should pay more attention to in the future[2].

[2] This attention-focusing mechanism is similar in spirit to the reinforcement learning techniques known as prioritized sweeping [MA93] and Queue-Dyna [PW93].

Chapter 5

Conclusion

This thesis has addressed the incorporation of a low-level cognitive ability into reactive, behavior-based artificial intelligence architectures. Specifically, it discussed the need to generate short-term, observation-based expectations about the world and react appropriately to the violation of those expectations.

The motivation for incorporating expectations was discussed and a theory of expectations and expectation violations was proposed. To test that theory it was implemented as a non-intrusive extension to an existing reactive behavior-based architecture; this implementation proved successful in replicating many of the phenomena predicted by the theory.

In conclusion, I believe that this model was successful in helping behavior-based creatures compensate for the limitations of perception, thereby facilitating behavior whose adequacy, coherence, and relevance is qualitatively closer to that of real animals.

Bibliography

[AC87] Philip Agre and David Chapman. Pengi: An implementation of a theory of activity. In *Proceedings AAAI-87*, San Mateo, CA, 1987. Morgan Kaufmann.

[Bae76] G. Baerends. On drive, conflict and instinct, and the functional organization of behavior. *Perspectives in Brain Research*, 45, 1976.

[Blu94] Bruce Blumberg. Action-selection in hamsterdam: Lessons from ethology. In *From Animals to Animats: Proceedings of the Third International Conference on the Simulation of Adaptative Behavior*, MIT Press, Cambridge MA, 1994.

[Blu96] Bruce Blumberg. *Old Tricks, New Dogs: Ethology and Interactive Creatures.* PhD thesis, The Media Laboratory, Massachusetts Institute of Technology, September 1996.

[Bre98] Cynthia Breazeal. A motivational system for regulating human-robot interaction. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pages 54–61, Menlo Park, July 26–30 1998. AAAI Press.

[Bro86] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, April 1986.

[Bro91a] Rodney A. Brooks. Challenges for complete creature architectures. In Jean-Arcady Meyer and Stewart W. Wilson, editors, *From animals to animats*, pages 434–443. First International Conference on Simulation of Adaptive Behavior, 1991.

[Bro91b] Rodney A. Brooks. Intelligence without reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 569–595, Sydney, Australia, August 1991. Morgan Kaufmann.

[Bro96] Rodney A. Brooks. Prospects for human level intelligence for humanoid robots. In *Proceedings of the First International Symposium on Humanoid Robots (HURO-96)*, 1996.

[Den87] Daniel C. Dennet. *The Intentional Stance*. MIT Press, Cambridge, MA, 1987.

[Den98] Daniel C. Dennet. *Brainchildren: Essays on Designing Minds*. MIT Press, Cambridge, MA, 1998.

[Dre91] Gary L. Drescher. *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. Artificial Intelligence Series. MIT Press, Cambridge, MA, 1991.

[Fir87] R. James Firby. An investigation into reactive planning in complex domains. In Howard Forbus, Kenneth; Shrobe, editor, *Proceedings of the 6th National Conference on Artificial Intelligence*, pages 202–206, Seattle, WA, July 1987. Morgan Kaufmann.

[Gel74] A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.

[Hau98] Marc D. Hauser. Expectations about object motion and destination: Experiments with a nonhuman primate. *Developmental Science*, 1(1):31–38, 1998.

[JWB+99] Michael P. Johnson, Andrew Wilson, Bruce Blumberg, Christpher Kline, and Aaron Bobick. Sympathetic interfaces: Using a plush toy to direct synthetic characters. In *SIGCHI'99*, pages 152–158, Pittsburgh, 1999. ACM Press.

[Kru62] Leonid Viktorovich Krushinskii. *Animal Behavior: Its Normal and Abnormal Development*. International Behavioral Sciences Series. Consultants Bureau, New York, 1962. Translated from the original Russian text published in 1960 by Moscow University Press.

[LeD96] Joseph LeDoux. *The Emotional Brain*. Simon and Schuster, New York, 1996.

[Lor73] Konrad Lorenz. *Foundations of Ethology*. Springer-Verlag, New York, 1973.

[Lud76] A. Ludlow. The behavior of a model animal. *Journal of Behavior*, 58, 1976.

[MA93] Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*, 13, 1993.

[Mae90] Pattie Maes. How to do the right thing. *Connection Science Journal*, 1(3):293–325, 1990.

[Mat94] Maya Mataric. Interaction and intelligent behavior. MIT Artificial Intelligence Laboratory Technical Report AI-TR-1495, Massachusetts Institute of Technology, 1994.

[McF73] David McFarland. *Animal Behavior*. Longman Scientific and Technical, Harlow, UK, 1973.

[MDBP94] Pattie Maes, Trevor Darrell, Bruce Blumberg, and Alex Pentland. The ALIVE system: full-body interaction with animated autonomous agents. MIT Media Lab Perceptual Computing Group Technical Report No. 257, Massachusetts Institute of Technology, 1994.

[MDBP96] Pattie Maes, Trevor Darrell, Bruce Blumberg, and Alex Pentland. The ALIVE system: Wireless, full-body interaction with autonomous agents. *ACM Special Issue on Multimedia and Multisensory Virtual Worlds*, 1996.

[MH69] John McCarthy and P. J. Hayes. Some philosphical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*. Edinburgh University Press, Edinburgh, 1969.

[Min86] Marvin Minsky. *The Society of Mind*. Simon and Schuster, New York, 1986.

[Pia52] Jean Piaget. *The Origins of Intelligence in Children*. Norton, New York, 1952.

[Pia54] Jean Piaget. *The Construction of Reality in the Child*. Norton, New York, 1954.

[PW93] Jing Peng and Ronald J. Williams. Efficient learning and planning within the Dyna framework. *Adaptive Behavior*, 1(4):437–454, 1993.

[RCB98] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–41, September/October 1998.

[Rei96] W. Scott Reilly. *Believable Social and Emotional Agents*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1996.

[Rey87] Craig Reynolds. Flocks, herds, and schools: A distributed behavior model. In *Proceedings of SIGGRAPH 87*, 1987.

[Spe85] E. S. Spelke. Preferential looking methods as tools for the study of cognition in infancy. In G. Gottlieb and N. Krasnegor, editors, *Measurement of Audition and Vision in the First Year of Postnatal Life*, pages 85–168. Ablex, Norwood, NJ, 1985.

[TJ81] Frank Thomas and Ollie Johnson. *The Illusion of Life: Disney Animation*. Hyperion, New York, 1981.

[TT94] Xiaoyuan Tu and Demetri Terzopolous. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of SIGGRAPH 94*, 1994.

[Tyr93] Toby Tyrrell. *Computational Mechanisms for Action Selection*. PhD thesis, Centre for Cognitive Science, University of Edinburgh, 1993.

[Vel98] Juan Velasquez. When robots weep: Emotional memories and decision-making. In *Proceedings of AAAI 98*, 1998.

[WB95] Greg Welch and Gary Bishop. An introduction to the Kalman filter. Computer science technical report TR-95-041, University of North Carolina at Chapel Hill, 1995.

[WB99] Andrew D. Wilson and Aaron. F. Bobick. Realtime online adaptive gesture recognition. MIT Media Lab Perceptual Computing Group Technical Report 505, Massachusetts Institute of Technology, 1999.