It's about Time: Temporal Representations for Synthetic Characters

by

Robert Carrington Burke

B. Sc. Eng., Queen's University at Kingston, 1999

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 2001
Revised September 2001

Signature of Author .....
Program in Media Arts and Sciences
September 4, 2001

Certified By .....
Bruce M. Blumberg
Associate Professor of Media Arts and Sciences
MIT Media Laboratory
Thesis Supervisor

Accepted By .....
Stephen A. Benton
Chair, Departmental Committee of Graduate Students
Program in Media Arts and Sciences

It's about Time: Temporal Representations for Synthetic Characters

by

Robert Carrington Burke

B. Sc. Eng., Queen's University at Kingston, 1999

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

Abstract

Inspired by recent work in ethology and animal training, we integrate representations for time and rate into a behavior-based architecture for autonomous virtual creatures. The resulting computational model of affect and action selection allows these creatures to discover and refine their understanding of apparent temporal causality relationships which may or may not involve self-action. The fundamental action selection choice that a creature must make in order to satisfy its internal needs is whether to explore, react or exploit. In this architecture, that choice is informed by an understanding of apparent temporal causality, the representation for which is integrated into the representation for action. The ability to accommodate changing ideas about causality allows the creature to exist in and adapt to a dynamic world. Not only is such a model suitable for computational systems, but its derivation from biological models suggests that it may also be useful for gaining a new perspective on learning in biological systems. The implementation of a complete character built using this architecture is able to reproduce a variety of conditioning phenomena, as well as learn using a training technique used with live animals.

It's about Time: Temporal Representations for Synthetic Characters

by

Robert Carrington Burke

The following people served as readers for this thesis:

Reader: .....

Whitman Richards
Professor
MIT Artificial Intelligence Laboratory

Reader: .....

Charles Ransom Gallistel
Professor
Center for Cognitive Science
Rutgers University

Table of Contents

Acknowledgements **11**

1.0 Introduction **13**

1.1 From a Cognitive Architectures perspective *13*

1.2 From an Animal Behavior perspective *14*

1.3 Thesis Overview *14*

2.0 Background: Sources of Inspiration **17**

2.1 But first, some Definitions *17*

2.2 Philosophical Inspiration: Synthetic Characters Group *18*

2.3 Architectural Inspiration: Synthetic Characters Cognitive Architecture *18*

2.3.1 Representing the World 19

2.3.2 Action Selection, Attention Selection 21

2.3.3 Locomotion and Interaction 22

2.3.4 Summary 22

2.4 Ethological Inspiration: Time, Rate and Conditioning *22*

2.4.1 Time, Rate and Conditioning Representations 23

2.5 Summary of Our Goal *26*

3.0 Cognitive Architecture **29**

3.1 Creatures must satisfy Internal Needs *29*

3.1.1 Drives represent Needs 29

3.1.2 Drives are satisfied by performing actions or consuming resources 31

3.1.3 Represent Drives with Various Levels of Abstraction 31

3.1.4 Affective Variables represent Emotional State 31

3.2 The fundamental action selection choice: explore, exploit or react. *32*

3.2.1 Overview 35

3.2.2 Exploit 35

3.2.3 Explore 36

3.2.4 React 36

3.2.5 Startle (Reflex actions) 37

3.2.6 Action Selection Summary 37

3.3 Apparent Temporal Causality: What, Why and How? *37*

3.3.1 First of all, we represent Stimuli 37

3.3.2 Predictors represent apparent temporal causality relationships 38

3.3.3 How to Learn Causality on the Job 40

3.3.4 The representation of action reflect ideas about causality 44

3.3.5 Changing ideas about causality 45

3.3.6 Forgetting must accompany learning 48

3.4 Apparent Temporal Causality lets us model the Effects on Affect *48*
3.4.1 Generating New Reward Markers 50

3.5 Summary *50*

4.0 Results for Synthetic Characters **53**

4.1 Integration into a complete Architecture *53*
4.1.1 Finding Stimuli in existing Perceptual Representations 54
4.1.2 A Separate Statistics and Filtering Agent 54
4.1.3 Roadmap for the rest of the Section 54

4.2 How Time Learning Affects Duncan *55*

4.3 The Goatzilla Domain *56*
4.3.1 Perception System 57
4.3.2 Autonomic Variable System 58
4.3.3 Action System 59
4.3.4 Navigation and Motor Control 60

4.4 Experiments: How This Works in Practice *60*
4.4.1 Learning about the World, and Recovering from Mistakes 60
4.4.2 Learning Curves for "Kicking Produces Food" Predictor 62
4.4.3 Clicker Training 65

4.5 Summary *67*

5.0 Results for Cognitive Psychology **69**

5.1 The Utility of Time Scale Invariance *69*

5.2 The Rate Estimation Challenge *69*

5.3 Different Answers to Basic Questions Redux *70*

5.4 Cue Competition *73*
5.4.1 Blocking 74
5.4.2 Overshadowing 74
5.4.3 One-Trial Overshadowing 75
5.4.4 Relative Validity 75
5.4.5 Inhibitory Conditioning 76

5.5 Background Conditioning *77*

5.6 Backward Conditioning *77*

5.7 Section summary *78*

6.0 Related Work **81**

6.1 Virtual Ethology *81*

6.2 Models of Conditioning *81*

7.0 Concluding Remarks **83**

7.1 Important Ideas *83*

7.1.1 Causality and Action Selection are integrated 83
7.1.2 Attention Selection and Action Selection are integrated 83
7.1.3 A Desire to Understand the World Drives Learning 83
7.1.4 The Cognitive Economy 84
7.1.5 Nothing is Deterministic, and Many Distributions aren't Linear 84
7.1.6 A Good Visualizer Is Worth Thousands of Lines of Debug Spew 85
7.1.7 The World Resists Oversimplification (Beyond Simple Credit Assignment) 85

7.2 Summary of Contributions 85
7.2.1 Implementation 85
7.2.2 Functionality 86

7.3 Future Work 86
7.3.1 Further integration of rate information 86
7.3.2 Integration of other Explanations 87
7.3.3 Grouping of ActionTuples 87
7.3.4 Long-Term Memory 87
7.3.5 Learning Higher-Level Goals and Concepts 87
7.3.6 Theory of Mind 87
7.3.7 Augmented Predictor Generation: Playing with Cause and Effect 88
7.3.8 Negative Knowledge and the Culling Sentinel 88
7.3.9 Spontaneous Recovery and the Culling Sentinel 88

8.0 List of References 91

Appendix A: Classification Techniques 95

Utterance Classification: DogEar 95

Gesture Classification 96

Appendix B: Mathematics 99

Scalar Expectancy Theory 99

Drives and DriveVectors 99

Utility and Affective Stance 100

Predictors: Reliability 100

Predictors: Trials and Interval Learning 101

ActionTuples 102

Action Selection 102

Acknowledgements

Two years ago, my advisor Bruce Blumberg took a chance on this wide-eyed Canadian kid who said he wanted to learn about learning. If this thesis tells a coherent story at all, it is because that kid has since then been influenced by Bruce's passion, perspective, smarts and focus. Now, with heartfelt thanks, he is off to spread the word.

In the meantime, I have had the opportunity to interact with the most intriguing cast of Characters. Ken Russell and Mike Hlavac welcomed me into this world. Song-Yee Yoon was my first sensei. Scott Eaton has been the source of the most enduring and endearing collaborations, and without his artistic talent there would be no Goatzillas in the mist. Damian Isla should already know that imitation is the greatest form of flattery, and through our work both architectural and theatrical I have held him in the highest regard. I will remember Michael Patrick Johnson's sage advice and savage squash arm. His comments on an earlier draft of this thesis improved it immensely. Yuri Ivanov helped give Duncan the Highland Terrier his ear, then filled our ears with acoustic guitar, and all the while offered me perspective. Chris Kline knew all along that prediction was the path we needed to walk. Marc Downie is the world's most endless source of ideas. Bill Tomlinson introduced me to the surf. Ben Resner understood better than any of us how to suck the marrow out of this place. Ari Benbasat and Spencer Lynn are both irreplaceable, but in very different ways. Aileen Kawabe somehow kept us afloat. Our UROPS – Matt Berlin, Jesse Gray, Jed Wahl and the others – did all the work. Dan Stiehl even breathed life into armature wire and silicone. And speaking of life, Gary McDarby and the crew at MediaLabEurope – Daragh McDonnell, Phil McDarby and Laura Nolan – helped spawn the most fearsome creature ever to be clicker trained.

My wonderful housemates – Mike Hlavac, Chris Kline, Natalie Brainerd and Miriam Korn – have patiently tolerated my manic lifestyle as I finish this thesis. Craig Calvert, Emma McCutcheon, Stephen Small and Matt Chilton offered me places to stay in faraway places while I hammered out a first draft.

My thesis readers have contributed tremendously to this thesis despite putting up with ludicrous deadlines. Whitman Richards made me think about representation, and asked me what I wanted our virtual creatures to do. Then Randy Gallistel offered me some representations, and showed what real creatures can do. I have been truly fortunate to work with these gentlemen and be guided by their ideas.

My unofficial thesis reader, Jessica Dowling, is a source of endless love and strength. My family – my parents Peter and Maureen, and my sisters Elizabeth and Caleigh – have put up with my total obsession with this work. I love these people more than any acknowledgements section can express. This thesis is dedicated to Grandmom, Mrs Dorothy Margaret Burke, who told me she wanted a Duncan of her own.

1.0 Introduction

In order to survive in a dynamic environment, many self-regulating systems – both biological and computational – make use of representations that model important aspects of the world. Two such representations fundamental for living systems are the passage of time, and the rate at which they experience relevant stimuli.

Early models of behavioral conditioning, such as the Rescorla-Wagner model, minimized the use of representation and speak simply of animals forming and strengthening associations between stimuli. While this associative model is successful at rendering explainable certain phenomena, there is a wide range of phenomena that it is unable to model without substantial trouble, such as the ability to learn an expected latency of reinforcement. Recent studies by Gallistel and others have considered the possibility that models of time and rate are fundamental to conditioning phenomena. Gallistel and Gibbon propose two new models – Scalar Expectancy Theory (SET) and Rate Estimation Theory (RET) – that require an animal to represent the length of the interval between stimuli, and the rate of reinforcement associated with various stimuli. Using these models, the authors are able to account for a number of conditioning phenomena that can not be explained using the Rescorla-Wagner model [Gallistel 1990], and they do so in a clear and elegant way.

Similarly, much of the early work in behavior-based artificial intelligence minimized the importance of representation [Brooks 1991b]. Recent work in the Synthetic Characters group involved incorporating time into the representations of a behavior-based system. The use of temporal representations was a bit ad-hoc, in that we used multiple representations spread throughout the system in a way that worked, but perhaps was not as elegant as one would wish. However, the use of time in the representation allowed us to model the kind of applied operant conditioning that underlies dog training. The sorts of learning that could occur within the previous framework included Thorndike's Law of Effect, wherein the relative frequency of behaviors reflects the relative value of their apparent consequences [Thorndike 1911]; cue learning, in which the system identified contexts in which particular actions are most reliable; behavioral shaping, in which the system learned the best way in which to perform a given action so as to improve its chances of desirable consequences; and the relative reliability of actions.

Our goal was to re-implement much of the learning mechanism in a way that pays attention to the sort of details that Gallistel attends to in the SET and RET models. The resulting representations and mechanisms needed to operate in real-time with dozens of potential stimuli. We wished to maintain, and hopefully improve upon, the system's ability to model a dog training paradigm and other sorts of learning.

We have arrived at the representations and mechanisms described in this thesis. They are not simply a recreation of Scalar Expectancy Theory and Rate Estimation Theory. Instead, they represent a hybrid that integrates new components inspired by Gallistel and Gibbon's work into the Synthetic Characters cognitive architecture. A creature constructed using this new architecture can predict and plan for future events by discovering causality relationships in the world. The creature is motivated to learn by a desire to explain salient stimuli it perceives. Its representation of apparent temporal causality is tightly integrated with its fundamental representation for action selection.

As we had hoped, the resulting architecture is capable of reproducing a wide variety of conditioning phenomena, as well as providing a robust implementation of the clicker training paradigm. We claim the following contributions based on the work presented in this thesis:

1.1 From a Cognitive Architectures perspective

- A model for action selection and learning that integrates apparent temporal causality into the action selection mechanism of a complete virtual creature.
- An implementation of that model's representations and mechanisms.

- Extensive visualizers that provide an observer with the ability to monitor the learning process.
- Two creatures that serve as robust, working examples.

1.2 From an Animal Behavior perspective

- The integration of computational models from ethology into a computational architecture designed to support non-trivial creatures in complex worlds.
- Discussion of how the behavior architecture is able to reproduce a variety of conditioning phenomena that can be observed in live subjects. Because it is derived from biological models, there is some suggestion that this architecture may allow us to gain a new perspective on learning in biological systems.
- Discussion of the benefits gained from being inspired by the ethological models, and the challenges discovered implementing the computational models described by SET and RET.

1.3 Thesis Overview

We begin in Section 2 with an introduction to the sources of inspiration behind this work: the philosophy of the Synthetic Characters group, the layered brain architecture upon which we built this system, and the ethology described in Time, Rate and Conditioning.

We then present in Section 3 a description of the new cognitive architecture. We begin with the notion that creatures have internal needs that they seek to satisfy, and show how this motivates their action selection decisions. We then demonstrate the benefits creatures obtain by learning about apparent temporal causality while they are on the job. We conclude with a discussion of how apparent temporal causality lets us model changes in a creature's affective state.

In Section 4, we present our results from a cognitive architectures point of view. We describe some lessons learned integrating the new representations and operations into the existing architecture, and then analyze two characters built using the new architecture. We use two scenarios – an applied operant conditioning technique, and another more whimsical domain in which a creature discovers apparent temporal causality in its world – to illustrate the learning process in more depth.

In Section 5, we discuss the results from an ethological point of view. We describe the challenges and successes we had integrating the SET and RET computational models into a complete cognitive architecture. We include a recreation of Gallistel and Gibbon's "Different Answers to Basic Questions" that contrasts the implemented model with the timing and associative models. We conclude by discussing the architecture's ability to reproduce a variety of conditioning phenomena that have been observed in real animals during experimental protocols.

In Section 6, we present references to some related work from both the virtual ethology and cognitive psychology fields.

We conclude in Section 7 with some of the important ideas that we uncovered during this research, followed by a summary of contributions and avenues for future research.

2.0 Background: Sources of Inspiration

This thesis begins with a discussion of the philosophy that inspires the Synthetic Characters group's attempt to understand the nature of intelligent behavior. This will lead us into an overview of the group's previous cognitive architecture.

Our goal is to augment that architecture to include a representation of apparent temporal causality. If we do this, a creature that uses the architecture will be able to use its *understanding* of past events, its *perceptions* of the present, as well as its *predictions* of future events, to inform decisions that help it satisfy its drives.

The search for appropriate representations leads us to our ethological inspiration. Scalar Expectancy Theory and Rate Estimation Theory provide a computational model that allows a creature to predict the timing of future events, and help decide which stimuli are worth attending to.

We will thus conclude the Section with our goal: to integrate elements of these ethological models into the cognitive architecture to provide the creature with an understanding of apparent temporal causality.

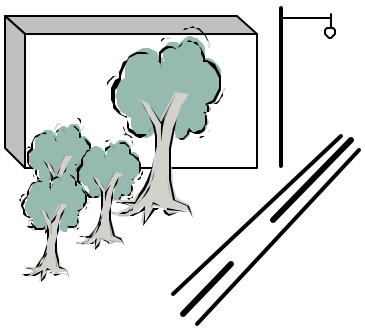
2.1 But first, some Definitions

Before we begin, we should define several important terms that we will use during this discussion: *representation*, *model*, *mechanism* and *architecture*.

A representational system has two essential ingredients:

- The *represented world*: that which is to be represented, and
- The *representing world*: a set of symbols or structures, each standing for something – or *representing* something – in the represented world.

Represented World



Representing World

Figure 1: The represented and representing worlds, after [Norman 1993] Figure 3.1.

In Figure 1, the "represented world" is shown on the left. The "representing world" is shown to its right as symbols on a sheet of paper. The representing world is an abstraction and a simplification of the represented world. In this example, the cube represents the building, and so on. The other aspects of the real (represented) world are absent from the representing world.

The power of a *representation* comes from its ability to help us understand the represented world. The choice of representation makes a dramatic difference in the ease of this task, even though, technically, the choice does not change the problem. A good representation allows us to *model* the important aspects of the represented world.

The proper choice of representation depends in part upon the *mechanism* being applied to the problem. The mechanism consists of procedures and other operations performed on the knowledge that is kept in the representations, and provides the method we will use to solve problems.

The *architecture* consists of both the representations and the mechanisms that work with them. The architecture thus encapsulates the complete problem-solving framework that we use *to represent and reason about the world.*

2.2 Philosophical Inspiration: Synthetic Characters Group

The Synthetic Characters group at the MIT Media Lab designs *cognitive architectures* for autonomous and semi-autonomous creatures that inhabit graphical worlds. By using ethological models to inform our design of these intelligent, expressive creatures, we seek to extend the work and philosophy formulated by Blumberg in [Blumberg 1996]. Previous extensions of the work have considered observation-based expectation generation [Kline 1999], the use of a character-based architecture for camera control [Tomlinson 1999], the use of classification techniques within the framework [Ivanov, Blumberg et al. 2000], extensions to characters' motor systems with applications to music [Downie 2001], and the use of quaternion-based animation blending techniques [Johnson 2001].

The group has recently sought insight into the nature of intelligent behavior by building characters inspired by the capabilities of dogs. While we still have a long way to go before we implement a perfect virtual dog, the trying has been very revealing. It has afforded us countless insights into the facets of intelligence we have yet to emulate, and required us to consider how to organize the many aspects of intelligence and behavior into a single cognitive architecture.

From a behaviorist's point of view, a focus on a particular species also provides us with a means for gauging our success. A problem with cartoon-like creatures is that they can behave however we want them to – there are no rules, and no way to ask "how close did we get?" At the other extreme, attempting to honestly model all aspects of human intelligence would be setting the bar too high (at this stage in the game). Later on in this thesis, we are able to examine how the behavior of one of our virtual creatures compares with the behavior of live subjects in a variety of experiments.

There is a vast amount of literature available on every conceivable aspect of canine existence (Fox examines social organization in wolves and other canines [Fox 1971]; Wilkes describes dog behavior and training for pet owners [Wilkes 1994]; Lindsay provides a thorough summary of dog behavior and training in two volumes [Lindsay 2001a], [Lindsay 2001b].)

Dog training is applied operant conditioning, and a domain in which one sees many of the phenomena described in lab experiments, but in the context of a whole "behaving" creature. One form of training, known as "clicker training," involves the use of a handheld device called the "clicker" that makes a short, sharp clicking noise. This noise serves as a precise event marker for the creature. When repeatedly followed by a treat, the noise of the clicker becomes associated with a food reinforcer. Clicker training has been successfully used to train animals ranging from dogs to dolphins (see [Pryor 1999] and [Wilkes 1995]). We will describe clicker training in more detail in Section 4.4.3, when we also demonstrate how the architecture described by this thesis allows us to perform clicker training on a virtual creature.

2.3 Architectural Inspiration: Synthetic Characters Cognitive Architecture

The Synthetic Characters group has designed and implemented an agent-based cognitive architecture that supports the creation of these virtual creatures. By agent-based we mean that the architecture is composed of many fairly simple components, each of which is individually unintelligent, but through their interaction they are capable of producing complex cognitive behavior. Most of the agents (or "Systems") contain their own representations and mechanisms.

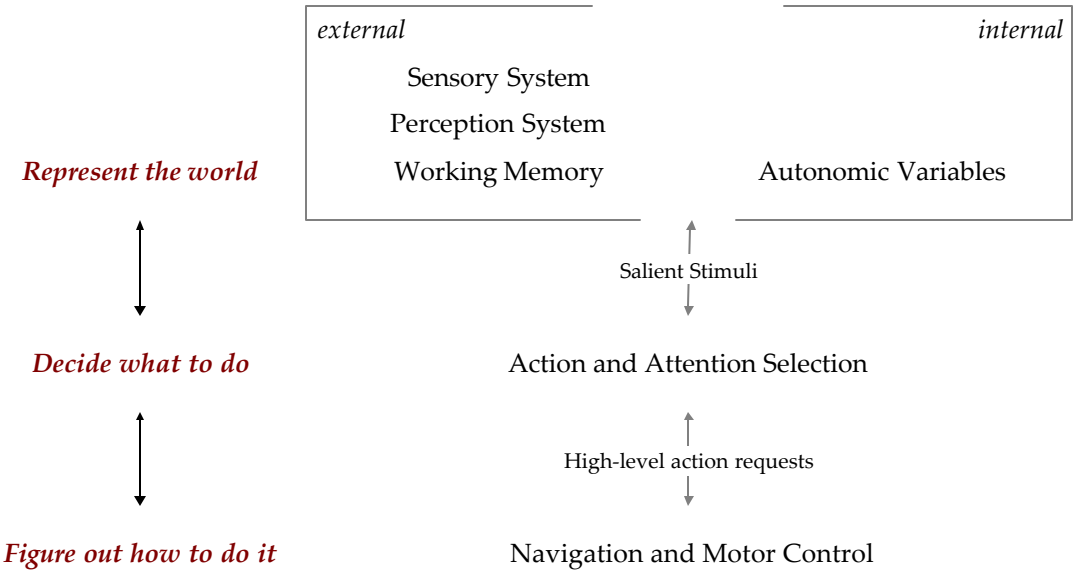


Figure 2: High-level view of the Synthetic Characters cognitive architecture.

We divide the systems in a creature's brain by function into three parts. As depicted in the Figure above, the first part of the cognitive architecture allows the creature to *represent the world*. The second part, which includes the action selection mechanism and its underlying representations, lets a creature *decide what to do*. The third part of the brain, which encapsulates navigation and motor control, helps a creature *figure out how to implement* its action plans.

No part of the architecture exists in isolation. The arrows in the Figure indicate bi-directional flow between its various components. Not only does our representation of the world help us decide what to do, but, in return, the action selection mechanism helps us refine how we represent the world. Similarly, not only do our navigation and motor systems carry out the high-level commands of our action selection mechanism, but through their operation we can learn things like how long it typically takes to perform an action.

2.3.1 Representing the World

The creature's ability to represent the external and internal worlds determines its ability to understand the current *context.* Many forms of learning are context-specific, in that they involve discovering important properties of different contexts. There are some contexts in which the creature should perform certain actions, others in which the creature can reliably predict future events, and yet others with which the creature associates an affective response. Thus, it is very important that we design representations that allow a creature to effectively determine its current context.

2.3.1.1 Sensing

A creature should only be able to act on information that its "sensory apparatus" is able to observe. The *Sensory System* marks the single entry point for external and proprioceptive sensory information into a creature's brain. All sensory input from the world is transformed into the creature's coordinate frame and filtered so that the creature can only receive "biologically plausible" sensory input. Thus, for a creature in a virtual world, the sensing mechanism acts as the enforcer of *sensory honesty.*

2.3.1.2 Perception

Once the sensory input has been "sensed," it can then be "perceived" by the creature's *Perception System*, which classifies and thus assigns meaning to every nugget of sensory information. The

distinction between sensing and perceiving is important. A creature, for example, may "sense" an acoustic event, but it is up to the perception system to recognize it as an instance of a specific type of acoustic event that has some meaning to the creature. When the shepherd says "down!" the sheep perceive it as just another human utterance, but the sheepdog interprets it as an acoustic pattern with a particular meaning.

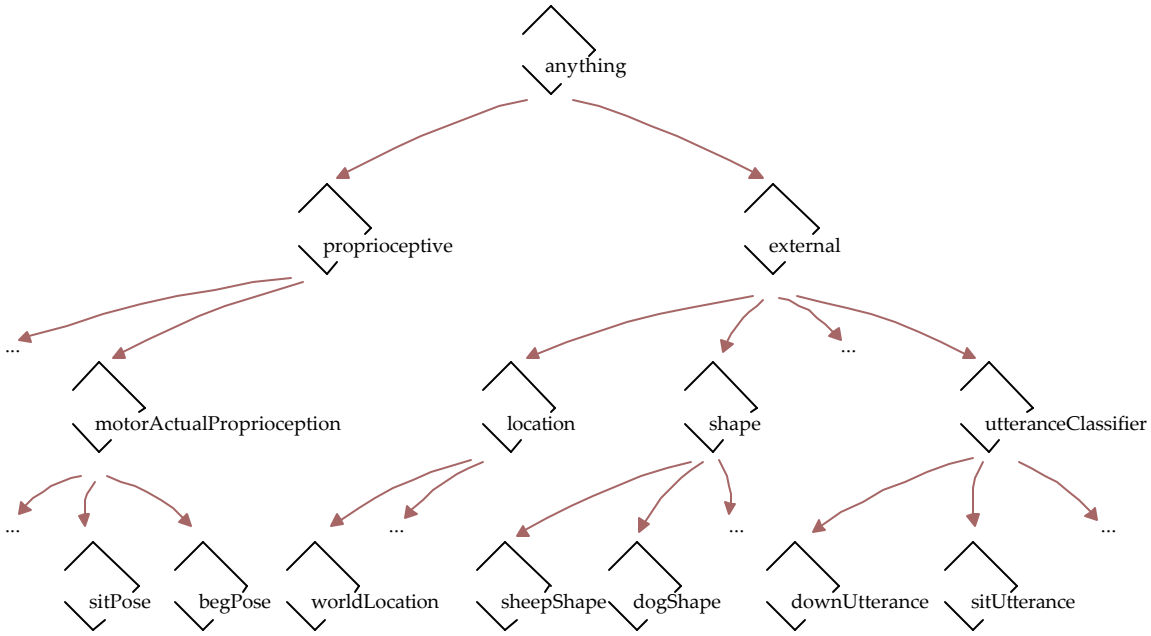


Figure 3: Part of a creature's Percept Tree.

As illustrated in Figure 3, creatures use a hierarchical arrangement of *Percepts*, each of which represents an atomic classification and data-extraction unit. Each Percept's *activation level* indicates its immediate or instantaneous response to sensory input. The *activation threshold* for each Percept (typically some small e) indicates the activation level above which it is considered *active*, and the *activation decay rate* indicates the rate at which the Percept's activation level decays in the absence of sensory input. In addition to these properties related to its activation, each Percept is attributed an *inherent salience*, a scalar metric that provides a common currency for salience across all Percepts.

The Percept provides a useful level of abstraction for reducing the dimensionality of incoming sensory information. An arbitrarily complex mechanism tucked into a Percept can determine whether or not it matches a particular sensory input. The current architecture contains examples of very simple matching mechanisms – for example, a string token mechanism for matching shapes in the virtual world – and also more complex mechanisms – for example, the acoustic pattern matcher, described in Appendix A, which interprets sound input.

Some Percepts are adaptive, using statistical models to characterize and refine their response properties. These Percepts can not only modulate their "receptive fields" (the space of inputs for which they will activate) but also, in concert with the action selection mechanism, *modify the topology of the Percept Tree itself*, producing an evolving hierarchy of children in a process called *perceptual innovation*. In general, Percepts are only prompted to perform perceptual innovation when the action-selection mechanism is fairly confident that this will allow the creature to make better decisions.

We emphasize here the importance of Percepts in determining the current *context*. Modifying the topology of the tree can allow the creature to represent different contexts in the world. Adding child

Percepts that represent increasingly specific perceptual responses lets the creature differentiate between increasingly specific *features* in the world.

2.3.1.3 Object Representation

Percepts reduce the dimensionality of incoming sensory information by transforming that information into *features* of the world that are meaningful to a creature. The creature also needs some sort of representation that matches features of differing modalities (visual, acoustic, etc.) by combining and identifying them together as an *object*. Such a mechanism would allow us to solve (or at least avoid) the perceptual binding problem, described by Treisman in [Treisman 1998].

We call our object representation a *Belief*. All of the Percept activity generated by each object in the world is stored together as a Belief in Working Memory. Each Belief, as depicted in Figure 4, consists of a history of each Percept's activation level as it pertains to that object in the world – in other words, it contains a history of *which features recently appeared in which objects*.

On timesteps in which no new information for a given Percept is observed, the confidence level of that Percept in the particular Belief is decayed. The rate of decay is determined in part by the Percept's activation decay rate. For example, confidence in another creature's location might decay rapidly without observation, but confidence in its shape probably should not.

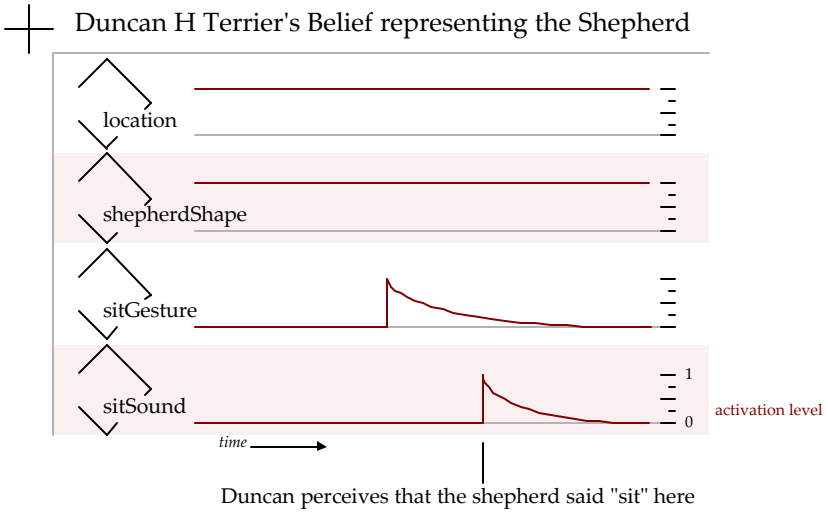


Figure 4: A Belief contains a history of how an object in the world activated each Percept.

An object representation allows us to query our memory in useful ways. Behaviors can be triggered by asking questions like, "is there food near me?" Action-targets can be picked by finding the "red object that is making the most noise." "Find an object that is humanoid-shaped and go to it" implies that you can extract both shape and location information out of a representation of an object. The Belief in Figure 4 is another creature's representation of the Shepherd. This creature, Duncan the terrier, believes the Shepherd just made the sit gesture, and then made an utterance that sounded like the word "sit."

2.3.2 Action Selection, Attention Selection

The action selection mechanism takes the creature's representations of the internal and external worlds, and uses the information they contain to arrive at a high-level plan that consists of three parts: a *desired action*, a *desired target for that action*, and the creature's current *focus of attention*.

2.3.3 Locomotion and Interaction

The creature carries out the high-level requests generated by the action selection mechanism by using its capacity for navigation and motor control to locomote around and interact with the world.

2.3.3.1 Separating Navigation from Action Selection

A separate Navigation System provides large-scale spatial competencies, usually by overriding motor requests passed down by the action selection mechanism to cause *locomotion* around the environment. This relieves the action selection mechanism of the burden of implementing the decisions it makes. Ultimately, the majority of animal behaviors follow the "approach, orient and do" model, and a separate navigation competence allows the action selection mechanism to concern itself with higher-level atoms.

2.3.3.2 Generating Expressive Motion

The Motor System is responsible for performing locomotion, a variety of actions, and orienting the eyes, heads, and bodies of creatures. Throughout all of a creature's motion, it should remain "in character," using cues from its motivational and affective state. In *The Illusion of Life*, Thomas and Johnson explain there is no such thing as *just* a "walk cycle." [Thomas, Johnson 1981] There is a happy walk, a sad walk, an excited walk, and so on. This notion – that an animation consists of both a *Verb* and an *Adverb* – is captured in the work of Rose et al [Rose, Cohen et al. 1999] that inspired our Motor System design. For more information about these and other Motor System issues, the reader is referred to [Rose, Cohen et al. 1999], [Downie 2001] and [Johnson 1999].

2.3.4 Summary

The Synthetic Characters architecture is very good at simulating virtual creatures that inhabit graphical worlds. The creatures are able to sense and perceive their worlds, and perform actions informed by those perceptions that help satisfy their drives (which we will discuss shortly in Section 3.1). They are even able to learn via a training technique based on applied operant conditioning. For the curious, a more detailed discussion of the layered brain architecture and training technique is found in [Isla, Burke et al. 2001], and even more implementation detail is found in [Burke, Isla et al. 2001].

One thing this architecture does not provide the creatures with is a representation of apparent temporal causality. If it did, creatures would be able to predict the onset of future events and thus plan for the future. Based on their understanding of apparent causality, they could perform actions explicitly intending to change the world in some way. They could expect an event at some future time, and react appropriately if that event does not occur.

We turn now to ethology, the study of animal behavior, for one model that may allow us to provide our creatures with some of this understanding.

2.4 Ethological Inspiration: Time, Rate and Conditioning

There are many reasons why we take classical ethology as an inspiration for the design of autonomous virtual creatures [Blumberg 1996]. By studying how animals behave and adapt in their natural environments, many ethologists like McFarland, Ludlow and Gallistel adopt the level of abstraction at which we would like to synthesize behavior (see [McFarland 1993], [Ludlow 1976], [Gallistel 1990]). Like ethologists, we are less concerned with how particular representations might be implemented at the level of neurons. Instead, like Minsky in his influential book Society of Mind [Minsky 1985], we are more concerned with how to organize and implement higher-level structures in the brain. As we saw in the previous subsection, we design systems and representations at the levels of perception, action selection, navigation and motor control. The interplay of many such simple systems might reproduce the wide variety of complex behavior that can be observed in nature.

2.4.1 Time, Rate and Conditioning Representations

Because of the success they had elegantly incorporating knowledge of time and rate into a computational model, we found inspiration for our implementation of apparent temporal causality in Gallistel and Gibbon's Time, Rate and Conditioning [Gallistel, Gibbon 2000]. In that article, the authors detail two theories that account for a broad range of conditioning phenomena. These theories depend on an animal's ability to learn temporal intervals between events, as well as rates of reinforcement. In Scalar Expectancy Theory, animals store in memory the reinforcement latency (the time between the onset of a stimulus and a subsequent reward signal). In Rate Estimation Theory, they store the rates of reinforcement for stimuli. The authors contrast their paradigms with the existing associative paradigms, and present a veritable library of experimental data to support their claims. What is exciting about the model is that by assuming the existence of representations for time and rate, Gallistel and Gibbon are able to easily explain a wide range of disparate conditioning phenomena.

2.4.1.1 Experimental Paradigm

One main goal of behaviorism is the identification of basic learning processes that can be described in terms of stimuli and responses. The experimental paradigm that underlies the study of conditioning is one in which the subject is presented with various stimuli. The subject learns associations between the stimuli. These associations often involve responses made by the animal when it perceives a stimulus.

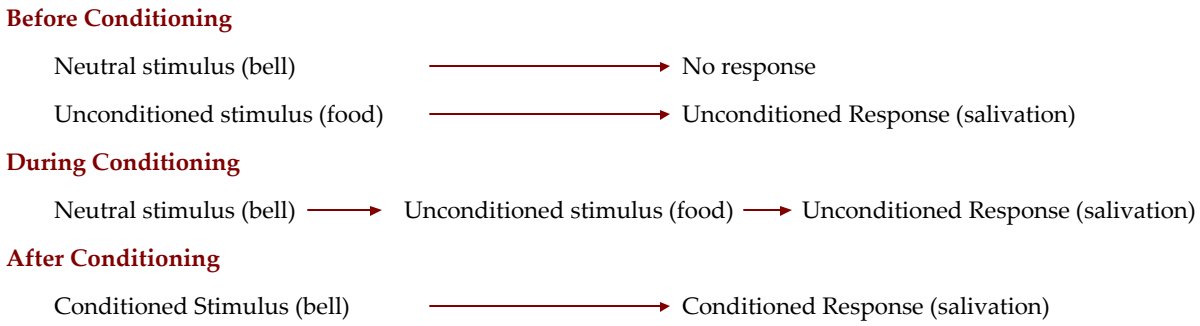


Figure 5: Classical conditioning procedure.

Pavlovian or Classical conditioning is an association-forming process by which a stimulus that previously did not elicit a response comes to elicit a response, in reflex-like fashion, after it is paired for one or more trials with a stimulus that already elicits a response. As shown in the above Figure, a neutral stimulus is initially demonstrated not to elicit a response. After it is paired for several trials with an *unconditioned stimulus* (abbreviated *US*), it becomes a *conditioned stimulus (CS)* and does elicit a *conditioned response (CR)*.

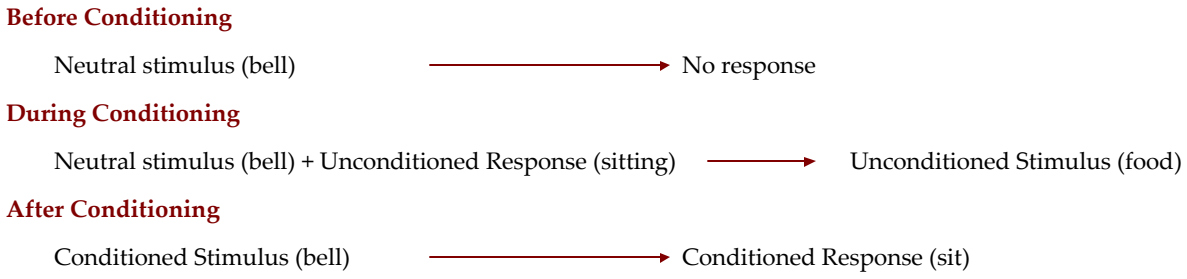


Figure 6: Operant conditioning procedure.

Operant or Instrumental conditioning is a process through which the consequences of a response increase or decrease the likelihood that the response will occur again. In one such procedure, the subject learns that performing a certain behavior in a context results in a reinforcer such as food. In accordance with Thorndike's Law of Effect, responses that produce a satisfying effect in a particular situation become more likely to occur again in that situation, and responses that produce a discomforting effect become less likely to occur again in that situation (see [Thorndike 1911]).

Most contemporary associative theorists no longer assume that the association-forming processes in classical and operant conditioning are fundamentally different. Rather, they are thought to give rise to different associative structures via a single association-forming process.

2.4.1.2 Scalar Expectancy Theory: the "When" decision

Scalar Expectancy Theory, or SET, pertains to the onset of the conditioned response (CR) following a stimulus onset, revealing both "when" and "for how long" the CR should occur. It explains how a subject's uncertainty about the "true value" of a remembered length of an interval is proportional to the length of that interval. The results produced by SET correlate with some well-established facts about how subjects time the duration between two events:

- First, the conditioned response (which suggests the expectation of the second event) is maximally likely at the reinforcement latency. When there is a fixed interval between two events – for example, a marking event and the delivery of a reinforcement – the probability that a well-trained subject will make a conditioned response *increases as the interval between events approaches, reaching a maximum at the interval length.*
- Second, the distribution of Conditioned Response onsets and offsets is scalar, and thus the temporal distribution of CR initiations and terminations is *time scale invariant.* In other words, when one signal seems to predict a future event, the approximate size of the window in which a subject expects that event to occur increases as the length of the interval between the events increases.

The components of the Scalar Expectancy Theory model include (1) a timing mechanism, (2) a memory mechanism, (3) sources of variability or noise in the decision variables, and (4) a comparison mechanism adapted to that noise. At the onset of the conditioned stimulus (CS), the *timing mechanism* generates a signal that is proportional at every moment to the elapsed duration of the subject's current exposure to the CS. This quantity represents the subject's measure of the duration of an elapsing interval. The timer is reset to zero by the onset of a reinforcer or other unconditioned stimulus (US) that marks the end of the interval. The magnitude of the timing signal at the time the timer is reset is written to *reference memory* through a multiplicative translation variable k*, whose expected value is close to but not identically one. This effect, known as *k* error*, causes the recorded interval to deviate from the timed value by some generally small percentage.

When the CS reappears (marking the beginning of a new trial), a new timing mechanism generates a signal, the subjective duration of which is compared to the remembered reinforcement delay in memory. The comparison takes the form of a ratio that is called the *decision variable*. When this ratio exceeds a threshold – $ß_1$ somewhat less than 1 – the animal responds to the CS, provided it has had sufficient experience with the CS to have already decided that it is a reliable predictor of the US. If the expected US does not occur, the conditioned response ceases to occur when the decision ratio exceeds a second threshold $ß_2$ somewhat greater than 1. In other words, the subject begins to respond when it estimates that the currently elapsing interval is close to the remembered interval. If the US does not appear, the subject stops responding when it estimates that the currently elapsing interval is sufficiently past the remembered interval. The ß decision thresholds constitute the criteria for "close" and "past." The measure of closeness is the ratio between the currently elapsing interval, and the remembered interval.

A diagram might help here.

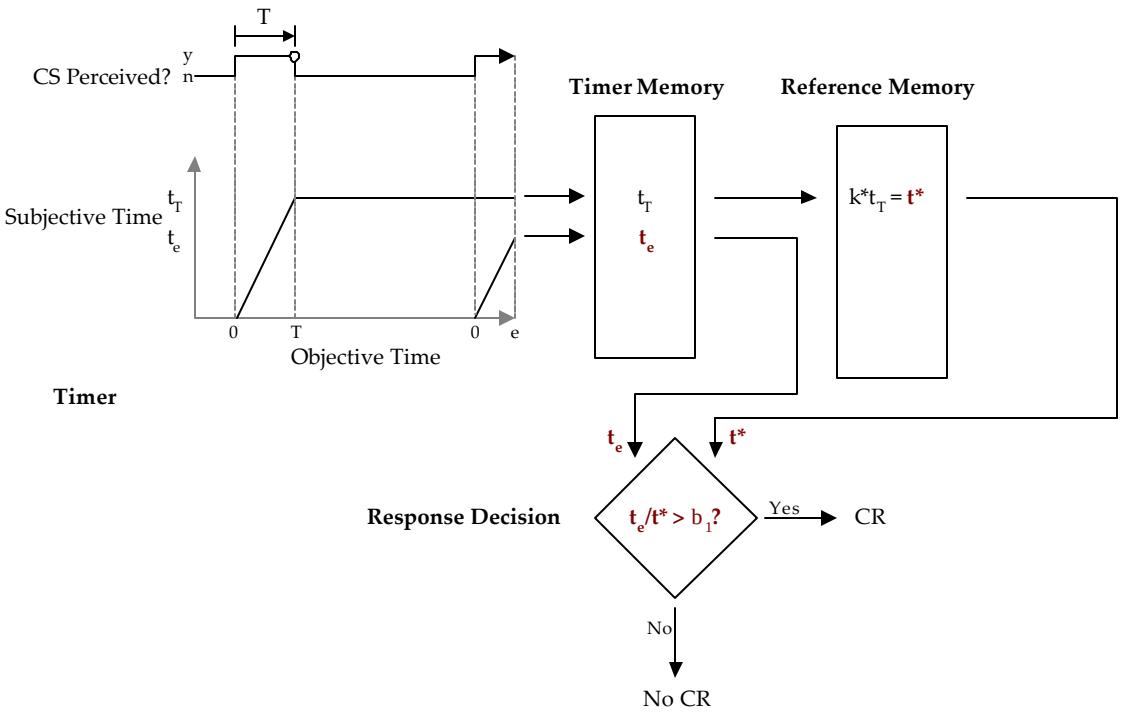


Figure 7: Flow diagram for *Scalar Expectancy Theory*, explaining the timing of an animal's conditioned response. After [Gallistel, Gibbon 2000], Figure 3.

In Figure 7, we see how Scalar Expectancy Theory handles two trials. Each trial involves the activation of a timing mechanism (left side of Figure). The first trial was reinforced at time T (the circle on the timeline), and the second trial is still elapsing at time e. When the first trial was reinforced, the *cumulated subjective time*, t_T, was stored in Timer Memory and transferred to Reference Memory via a multiplicative variable k*, thus producing the so-called k* error and encoding the remembered interval $t^*=k^*t_T$. The subject decides when to respond by using the ratio of the elapsing interval in Timer Memory (t_e) to the remembered interval in reference Memory (t^*). When the ratio exceeds a threshold, $ß_1$, which is close to but generally less than 1, the subject responds.

To summarize, Scalar Estimation Theory employs two assumptions to explain scale invariance in the distribution of conditioned responses:

- The decision variable used to determine when to respond takes the form of a ratio t_e/t^*, the denominator of which, t_e, is the learned interval length; and the numerator of which, t^*, is the elapsed time since the conditioned stimulus.
- Estimates of duration read from memory have scalar variability.

2.4.1.3 Rate Estimation Theory: the "Whether" decision

Scalar Estimation Theory assumes that the animal has already determined whether or not a stimulus merits a response. In the Rate Estimation Theory model, this decision is based on an animal's growing certainty that a stimulus has a substantial effect *on the rate of reinforcement*. In simple conditioning, this appears to be determined "by the subject's estimate of the maximum possible value of the rate of background reinforcement given its experience of the background up to a given point in conditioning." [Gallistel, Gibbon 2000] Gallistel and Gibbon provide a computational model for how animals determine the true rates of reinforcement for each stimulus and use this to determine whether or not a

stimulus merits response. They demonstrate how this model accounts for experiments that employ fixed and variable rates of reinforcement.

Conditioning to one stimulus does not proceed independently of conditioning that occurs to other stimuli. Rate Estimation Theory provides an explanation for cue competition phenomena based on two principles: *rate additivity* and the *principle of parsimony.*

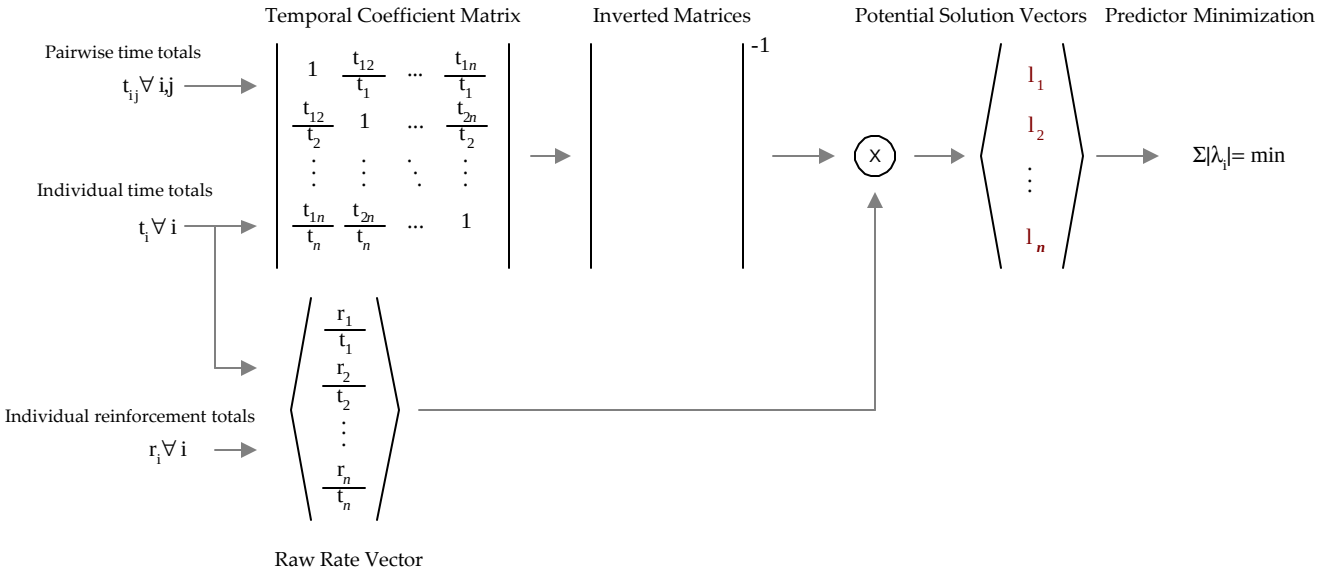


Figure 8: Functional structure of the computational process that underlies Rate Estimation Theory. After [Gallistel, Gibbon 2000], Figure 18.

For each stimulus, the subject stores *individual time totals* (t_i) *and pairwise time totals* (t_{ij}) in the *temporal coefficient matrix* (as shown in Figure 8). The *raw rate vector* consists of the rate estimates made by ignoring other stimuli and simply dividing the cumulative exposure to each stimulus (t_i) by the number of reinforcements obtained in the presence of that stimulus (r_i).

The creature arrives at the *corrected* or *true rates of reinforcement* ($?_i$) by inverting the temporal coefficient matrix and multiplying the inverse by the raw rate vector. If there are redundant stimuli, the determinant of the temporal coefficient matrix will be 0 and thus its inverse undefined. In this case, redundant stimuli are removed from the matrix to create lower-order matrices representing systems of equations that ignore one or more stimuli. The principle of predictor minimization (or more generally, the *principle of parsimony*) determines which of the lower-order solutions is taken as the "correct" solution: it is the solution that minimizes the sum of the absolute values of the predicted rates. Thus the creature arrives at the *true rate of reinforcement* for stimuli.

Rate additivity is implicit in the structure of the mechanism used by Rate Estimation Theory to compute the rates of reinforcement that are credited to each of the experimentally manipulated stimuli. The principle of parsimony – essentially Occam's razor – is invoked for cases where the principle of rate additivity does not determine a unique solution to the rate estimation problem. Mathematical details of the partitioning model are available in the appendices of [Gallistel, Gibbon 2000].

2.5 Summary of Our Goal

The Synthetic Characters cognitive architecture is very good at integrating several kinds of learning into an autonomous virtual creature that can often maintain the illusion of life. Scalar Expectancy Theory and Rate Estimation Theory succinctly explain the results of a wide variety of conditioning experiments. Perhaps most importantly, they explain how an animal can employ its ability to

remember the temporal interval between stimuli to predict a future event, and how an animal can decide which stimuli are worth responding to.

Our goal is to integrate aspects of Gallistel and Gibbon's computational model into the existing architecture. The resulting hybrid should allow us to build virtual creatures that are capable of learning apparent temporal causality relationships. The system will require a new action selection mechanism that allows the creature to take advantage of its understanding of causality.

The previous cognitive architecture could be said to integrate an analysis of the past with an ability to react to the present. With the new architecture, we seek to include an ability to predict the future. Thus a creature could be *informed* by salient stimuli perceived in the recent past, *reactive* to stimuli perceived in the present, and *able to plan* appropriately for the stimuli predicted to appear in the future.

We seek to preserve many of the qualities that make the existing system good, such as its modular nature, its capacity for intuitive behavior design, and its ability to reproduce operant conditioning phenomena. The new augmentations should further our ability to create robust creatures that are able to adapt to and learn in a dynamic environment.

3.0 Cognitive Architecture

By itself, a representation for apparent temporal causality won't improve the life of a virtual creature. We need to consider how a creature might *use* its knowledge of apparent causality to influence its action selection and help satisfy its internal needs.

Thus we begin this Section with the notion that a creature has *needs* that it must satisfy. The goal of the action selection mechanism is to explore, exploit and react to the world in a way that lets the creature satisfy those needs.

One thing that will help the creature perform these tasks better is the ability to learn from the past to predict the future. To allow this, we need to give the creature a means for predicting future events, which is something it would gain from an ethologically-inspired model of apparent temporal causality. We thus introduce the action selection mechanism and the representation of apparent temporal causality, and show how the two are integrated. We conclude the Section by showing how affective responses can be generated using our understanding of apparent temporal causality.

Mathematical details for many of the mechanisms described in this Section are found in Appendix B.

3.1 Creatures must satisfy Internal Needs

A creature's needs are a subset of the internal state that we need to represent. Our atomic component of internal representation is the *Autonomic Variable*. Autonomic Variables each produce a continuous scalar-valued quantity. Most Autonomic Variables have *drift points* – values that they drift toward in the absence of any other input.

3.1.1 Drives represent Needs

Some of the creature's Autonomic Variables represent *Drives*, like the *hunger* drive depicted in the Figure below. In addition to its drift point, each Drive also has a *set point*, the value at which the drive is considered satisfied. The strength of the drive is proportional to the magnitude of the difference between the set point and the output value.

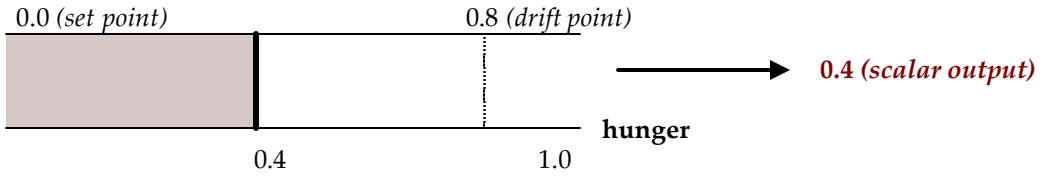


Figure 9: An Autonomic Variable, the atomic component of internal representation.

Associated with each Drive is a scalar *drive multiplier* that allows the creature to compare the importance of various drives. Over the course of a creature's existence, these multipliers might change, so that the creature can favor different drives at different times. This mechanism can be used to create periodic changes in the creature's drives (for example, to produce a circadian rhythm) and induce drive-based developmental growth over a creature's lifespan.

Take the output of all the Autonomic Variables that represent Drives, and concatenate their scalar output values into a vector, and we have the *DriveVector* – a summary of the creature's current drive state. As depicted in the following Figure, each component of the DriveVector indicates the state of a particular Drive, and the entire DriveVector summarizes the creature's current needs.

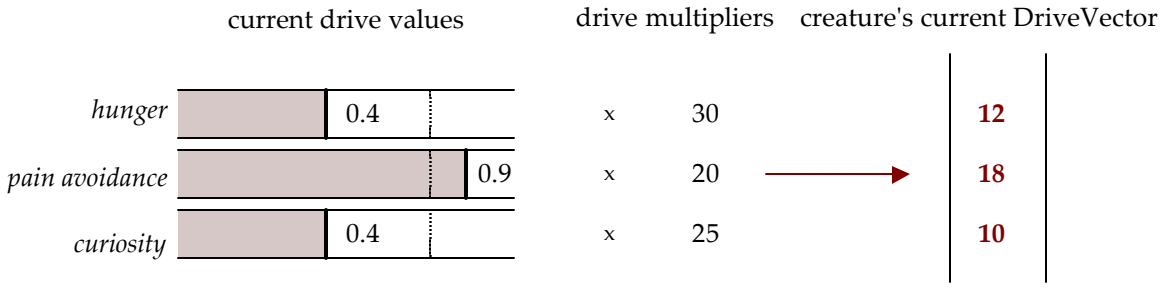


Figure 10: Three drives, their drive multipliers, and the resulting DriveVector.

This has ramifications on the way our creatures represent "value" or "goodness." Many machine learning algorithms use a single-dimensional "utility" value that describes something's general goodness. But by itself, an "affective tag" like this does not reflect how something's utility changes as the creature's drives change.

Instead, our creatures represent the value of something in the world – whether it is an action, a fellow creature, or an object – as a "value vector" with the same dimensions as the DriveVector. That vector indicates how the creature believes that thing will affect its drives.

The utility of something in the world at a given moment can be reduced to a scalar value by taking the dot product of the thing's value vector with the creature's DriveVector, as shown in Figure 11. The result of this approach, which parallels the motivational model described by Spier in [Spier 1997], is that the utility of something in the creature's world reflects the creature's current drive state. Please note that the *set point for each Drive in this discussion is zero.* Thus, *positive drive output values* indicate the magnitude of the creature's drive away from a zero set point. *Negative utility values* indicate something useful to the creature, because that thing has the perceived effect of *reducing the creature's drives*.

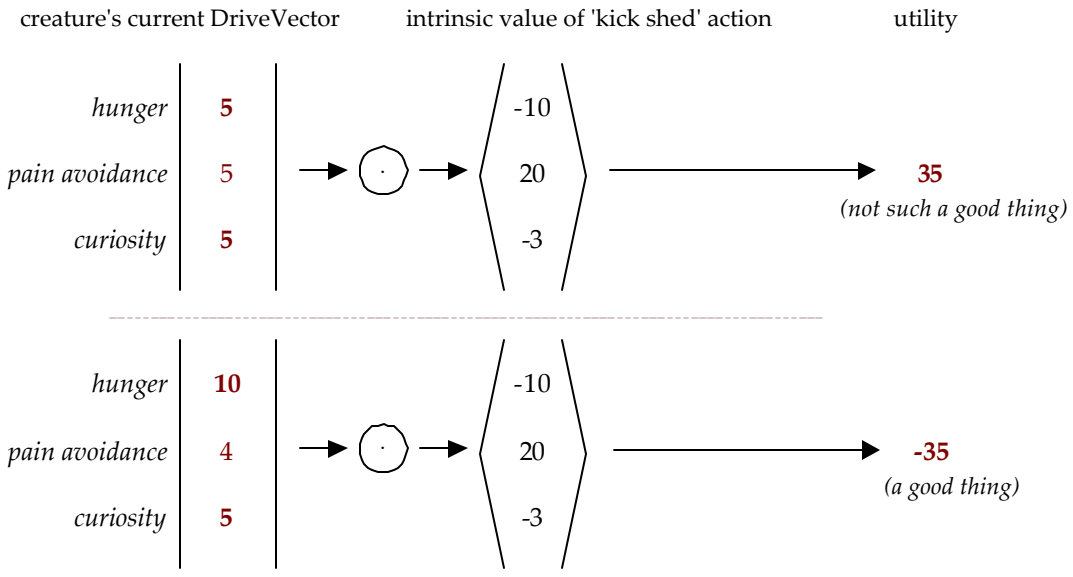


Figure 11: The utility of an action varies with the creature's current DriveVector.

The creature described in Figure 11 has three drives: hunger, pain avoidance, and curiosity. These are concatenated into a three-dimensional DriveVector $|d_1\ d_2\ d_3|$ (left side of Figure). The creature's food source is a shed in which there are sleeping sheep. If he rattles the shed, the sheep will scatter and he can feast. However, the shed is surrounded by an electrified fence. Thus, in order to rattle the shed, the

creature will have to sustain a shock, which will hurt a whole lot. Thus, the value of the "kick the shed" action (middle of Figure) might look like [-10 20 -3] relative to his drives [hunger, pain, curiosity], meaning that it will reduce his hunger drive (good), increase his pain (bad), and slightly lower his curiosity drive (because kicking stuff is intriguing). If this unnamed creature's current drives are [**5 5** 5], then the value of kicking the shed is [**5 5** 5] · [-10 20 -3] or 35, a positive number suggesting that, overall, the action will not be such a good thing. But, in the absence of other food sources, the creature's drives might eventually drift to [**10 4** 5] for [hunger pain dominate]. Now he's hungrier and isn't in quite as much pain. The dot product of [**10 4** 5] and [-10 20 -3] generates a utility of -35; in other words, an effective strategy for satisfying the current drives. (We note that this example is functionally analogous to a more mundane experiment wherein a rat is presented with a lever, surrounded by an electrified floor pad, which causes food pellets to be dispensed when pressed.)

For the purposes of the action selection mechanism that follows, and as seen in this example, a special drive called *curiosity* is added to each of the creatures. Curiosity tends to drift up over time, and drop back down as the creature does interesting things. As we will see when we discuss action selection, treating curiosity as a first-class drive is very effective for producing exploratory behavior.

3.1.2 Drives are satisfied by performing actions or consuming resources

There are number of ways that we can model effects on a creature's drives. Lorenz and Leyhausen posit in [Lorenz, Leyhausen 1973] that creatures find it inherently satisfying to perform particular actions and consume resources. When either of those conditions are met (the creature consumes food, comes in contact with an electric shock, enters an action state it finds inherently rewarding, or whatever else), the values of the corresponding Autonomic Variables are automatically updated.

3.1.3 Represent Drives with Various Levels of Abstraction

It is possible to use Autonomic Variables to model higher- or lower-level abstractions of the creature's internal state. For example, we could model a low-level representation of "chemicals" that are added to a creature's "bloodstream," much like the system used to create Cyberlife's Creatures [Cyberlife 1998]. Or, Autonomic Variables can model higher-level concepts, such as the desire to dominate other creatures, by generating an Autonomic Variable that represents the result of a function of other Autonomic Variables. A learning mechanism installed here would allow us to emulate the intriguing technique employed by the titans in Lionhead's Black and White, who use a form of perceptron training to learn which drives should propel them to pursue which consummatory actions [Evans 2001].

3.1.4 Affective Variables represent Emotional State

Autonomic Variables are also used to model the creature's emotional state. We have worked with several models of affect that create a multidimensional "affective space." Each axis of the space is represented by an Autonomic Variable we call an *Affective Variable*. Yoon describes the three-axis stance-valence-arousal model in [Yoon, Blumberg et al. 2000] that was inspired by Russell [Russell 1980]. Outputs from these axes can be mapped into less rudimentary affective states, such as the seven primary emotional states suggested by Ekman in [Ekman 1982]: surprise, interest, anger, disgust/contempt, happiness, sadness and fear.

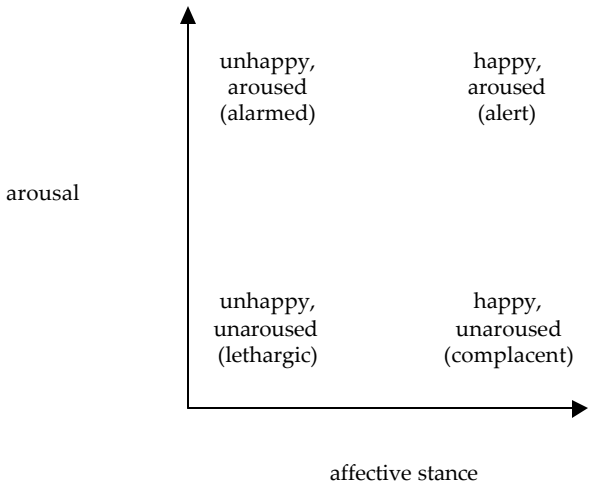


Figure 12: Simple two-axis emotional space, after [Russell 1980].

The creatures described in this document use the two-axis emotional space shown in Figure 12 which consists of an *arousal* axis, and an *affective stance* axis that integrates aspects of Yoon's stance and valence, restoring a two-axis model very similar to that proposed by Russell. (Russell calls the affective stance axis "pleasure" in [Russell 1980].)

3.2 The fundamental action selection choice: explore, exploit or react.

Now that we've established how creatures represent the needs that they must satisfy, we can discuss the action selection mechanism that helps them satisfy those needs. The fundamental choice a creature must make at every moment is whether to *exploit* its knowledge about the world, *explore* the world to possibly discover new things, or *react* to recently-observed stimuli.

The action selection mechanism that will integrate these *explore, exploit* and *react* operations should exhibit the qualities suggested by Brooks in [Brooks 1991a]. Every action performed by the creature should appear (and be) *relevant*. It should make sense, given the creature's internal state, its perceptions of its environment, its knowledge of how the world works, and its repertoire of actions. The creature's behavior should have a high degree of *persistence* and *coherence*, in that the creature should be aware of the appropriate duration of its actions and see them through to completion, without getting stuck in "mindless loops." The selection mechanism itself should be capable of *learning and adaptation*, and facilitate learning in other parts of the creature's brain.

Our task is complicated by the fact that the action selection mechanism we want is not purely reactive. We would like the mechanism to be *informed* by salient stimuli perceived in the recent past, *reactive* to stimuli perceived in the present, and *able to plan* appropriately for stimuli predicted to appear in the future.

We need a representation that can integrate the past, present and future, offering a creature an understanding of the passage of time.

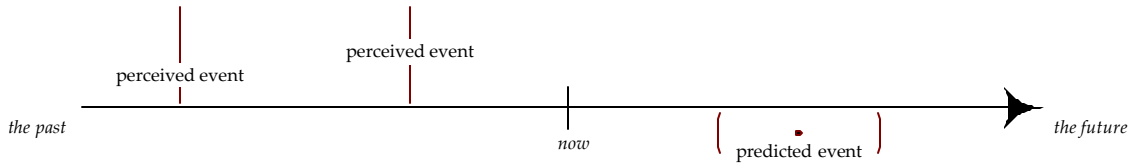


Figure 13: The TimeLine representation integrates past, present and future events.

This representation is the *TimeLine.* The creature uses it to maintain a list of salient events – both those *perceived* in the past and those predicted for the future – arranged on a temporal axis. The observation of a salient stimulus causes a *Perception event* to be posted to the TimeLine. In response to Perception events, a creature can use its understanding of cause and effect to add *Prediction events* to the TimeLine. A Prediction event includes the stimulus that is predicted, the time window in which the onset is predicted to occur, and the Predictor (discussed below) that caused the prediction to appear.

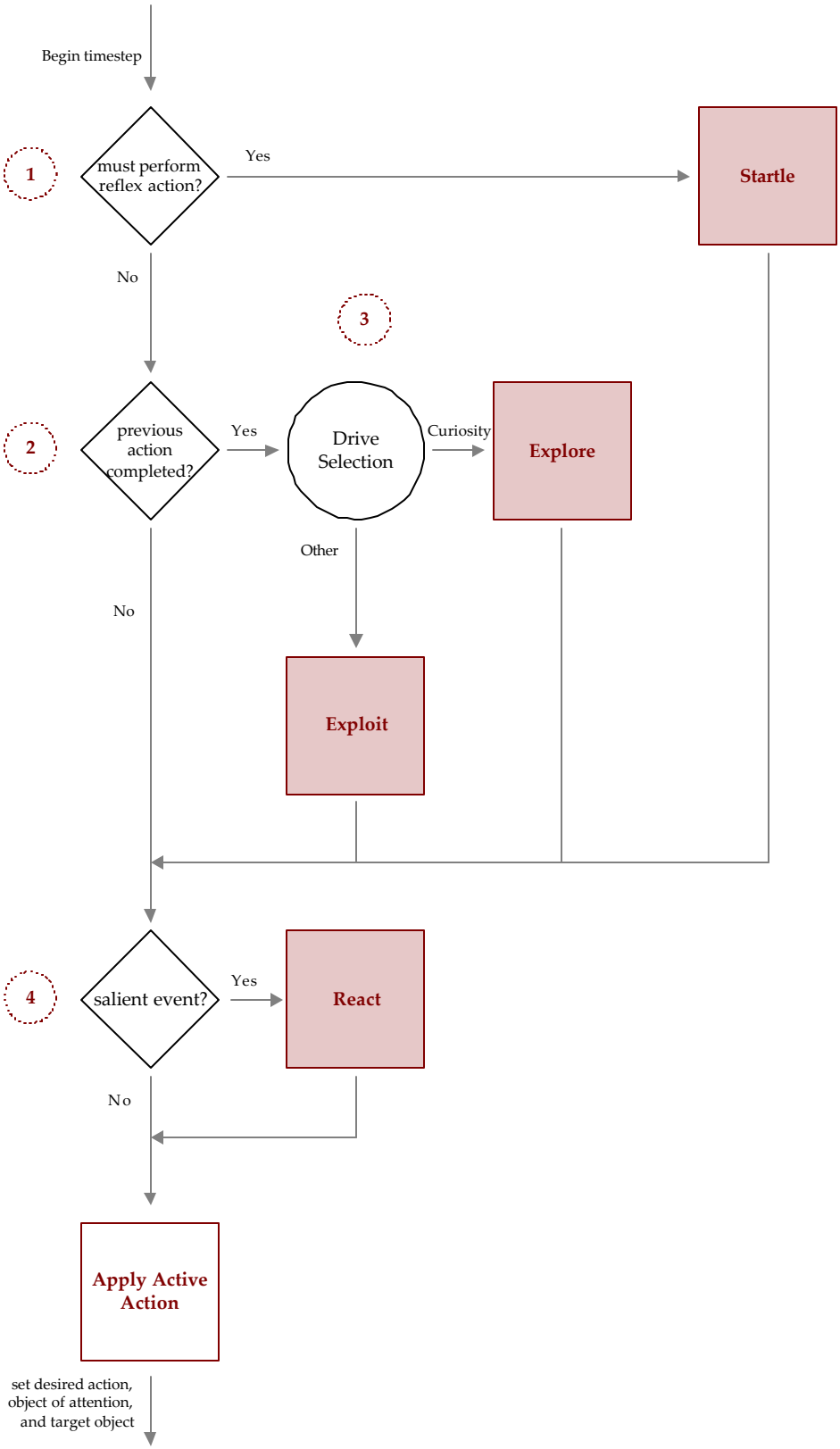


Figure 14: How the action selection mechanism integrates the explore, exploit, react and startle operations.

3.2.1 Overview

The previous Figure illustrates how the action selection mechanism integrates the explore, exploit, react and startle operations. On every timestep, we first check if the creature needs to perform a reflex or *startle* action *(**(1)** in the diagram)*. If not, we check if the active action is completed **(2)**. If so, the creature selects a drive on the basis of their relative magnitudes **(3)**. If the curiosity drive is chosen, the creature performs an *explore* operation. If any other drive is chosen (or the explore operation fails to select a new action), the creature performs an *exploit* operation, which is guaranteed to select a new desired action. Next, the *react* operation is performed on any newly salient stimuli, potentially causing the focus of attention and desired action to change **(4)**.

At the end of the timestep, the mechanism has in fact made three selections: it has chosen the *desired action*, the *object of attention*, and the *target object*. The *desired action* is a high-level token like "sit," "kick" or "approach" that describes what the creature would like to do. The *target object* is the object on which the desired action should be performed. The *object of attention* represents the creature's focus of attention. Each of these three selections is "winner take all," in that they are made to the exclusion of all other options for this timestep.

An example should illustrate the difference between the target object and object of attention. Suppose the creature is a dog that is running around a sheep. Then both his target object and his object of attention would be the sheep around which he's running. But further suppose his shepherd is shouting "Sit! Sit! C'mon boy, *sit!*" which he is choosing to ignore. This belligerent canine might acknowledge the (increasingly frenetic) vocalization by setting his object of attention to his master, thus causing his Motor System to dart a glance over his shoulder in the direction of the shouting. But because his target object remains the sheep, he'll still run around the sheep (and not start running around the shepherd).

We now describe the exploit, explore and reaction operations in more detail. These operations are supported by three special action states: *approach*, *avoid*, and *observe*, that represent different reactions a creature might have to a stimulus (either perceived or predicted).

Once again, although we are light on mathematical details in this section, a summary of operations is found in Appendix B.

3.2.2 Exploit

The *exploit operation* causes the creature to use its knowledge about the world to select an action it believes will help satisfy its drives. This may mean performing a consummatory action, or performing an appetitive action that the creature predicts will move it closer to performing a consummatory action.

The creature can exploit by using its direct perceptions of the world to choose a new action state with a high utility.

Or, the creature can exploit by reacting to something it predicts is about to happen. If a painful stimulus is almost certainly about to appear, it should be *avoided* if at all possible. Similarly, if a stimulus about to appear will facilitate a consummatory action, the best course of action might be to *approach* the stimulus in preparation for its arrival. These represent *preventative and preparatory* action states.

One useful property of the drive-based utility metric described in Section 3.1 is that it can be reduced to a scalar value that represents common value currency for different things in the world. In this case, we can use that currency to compare two kinds of action states: those triggered by perceptions, and preventative and preparatory action states that are triggered by predictions.

The scalar utility values obtained using both methods are used as input for a histogram probability distribution, from which the creature selects a single course of action in a winner-takes-all decision. A distribution is used here to heavily favor options with high magnitude values. The function used to map utility values to the histogram provides a useful degree of freedom representing "curiosity" that

can be used to tweak the creature's propensity to exploit the very best option. The spirit of this mechanism is to typically cause the creature to select the very best available option, while still occasionally selecting another option that seems very promising but not necessarily the "best."

3.2.3 Explore

There are many ways a creature can explore its world. It can redirect its attention toward an interesting object. It can explore that interesting object by performing actions on it; perhaps randomly, or perhaps by selecting actions that produced useful results for similar objects. It can select an action state because that state is interesting, rather than obviously useful. It can test predicting mechanisms in which it has low confidence, possibly by generalizing and discriminating the trigger contexts that cause them to make predictions. There are sufficiently many exploration techniques that, instead of peppering them throughout the action selection mechanism, we formalize our notion of exploration by encapsulating its many forms within the *explore operation*.

Like the exploit operation, the explore operation should end with the selection of the creature's next desired action state. However, unlike exploit, which culminates in a single action selection guided by a probability distribution, the explore operation requires at least two different selections. First, *Attention Selection* chooses an interesting target object for the creature. Next, *Drive Selection* chooses the drive that will guide exploration. Finally, depending on the result of these two selections, the creature may perform *Strategy Selection* to choose the exploration strategy it will use to select its next action state. Let's examine each of these Selections in more detail.

The process begins at *Attention Selection* with the creature choosing the object it will use as the target for exploration. Objects are selected on the basis of their "level of interest," an arbitrarily complex metric. In the current implementation, a creature's interest in an object tends to drift upward over time, decline when the object is the creature's focus of attention, and increase when unusual or salient perceptual activity is detected within the object. The creature's current object of attention is given preference, as are objects that are spatially proximate to the creature.

The creature now performs *Drive Selection* to decide what it's going to do with its newly acquired Object of Attention. A drive is selected probabilistically on the basis of its magnitude. If the drive selected is *not* the curiosity drive, the creature activates the previously-selected action state, setting the new Object of Attention to be its Target Object. As a result, the creature explores an interesting object in the world using existing techniques.

However, if Drive Selection returns the *curiosity* drive, the creature performs *Strategy Selection*, invoking more complex forms of exploration that possibly involve generating new action states. A number of strategies are possible here, including generalizing and discriminating the contexts in which the creature performs various actions.

3.2.4 React

The *react operation* gives the creature a chance to interrupt its current behavior and react to the perception of a salient stimulus.

The first thing a creature does when it perceives a salient stimulus – for example, a loud noise – is to try to explain it, by looking at the TimeLine for a prediction of the event. The creature's affective response will be largely determined by whether or not this event was predicted, and which action states, good or bad, the event will facilitate.

The creature may choose to interrupt its current behavior in favor of one that provides a reaction to the stimulus. Depending on the utility of the action states that the stimulus facilitates, the appropriate reaction may be to *approach*, *observe* or *avoid* it. Or, it may be to perform an action that is directly facilitated by the presence of this stimulus. The creature must decide whether to interrupt its current action to pursue one of these responses, again weighting the decision so that the currently active action will tend to persist unless a new option is significantly better.

If the creature can't explain the appearance of a stimulus, it is given an opportunity to invent an explanation for why it appeared, marking the beginnings of the apparent temporal causality process we will discuss in Section 3.3. For further discussion of "explaining away" unexpected events using probabilistic reasoning, see [Pearl 1988].

3.2.5 Startle (Reflex actions)

Sometimes, the creature must react to a stimulus with a *reflex action* more sudden and unstoppable than the mechanism provided by the react operation. The basic idea is that sometimes, due to constraints beyond the creature's control, the action selection mechanism must interrupt its current behavior to do something outside of behavioral control. A deafening noise unexpectedly occurring behind the creature should cause such a "startle" response. Similarly, coming in contact with a potent enough electric shock will cause the creature to convulse involuntarily. The action selection mechanism checks if it must initiate any such "startle " before performing one of the explore, exploit or react operations.

3.2.6 Action Selection Summary

The explore, exploit and react operations describe the approach the creature uses from moment to moment to explore the world, exploit knowledge about the world, and react to salient stimuli. In the next section, we will see how these operations guide the creature's attempts to learn how the world works.

3.3 Apparent Temporal Causality: What, Why and How?

The explore, exploit and react operations assume that the creature has the ability to represent *apparent temporal causality* relationships. But first things first: what do we *mean* by *apparent temporal causality relationships?* They are cause-and-effect relationships that a creature believes it has discovered in its world. They are *apparent,* because they are how the world appears to work to the character, whether or not the world actually works that way. They are *temporal,* because cause and effect are somehow related in time. And they represent *causality,* in that the creature can use them to generalize from specific examples to arrive at general principles about how the world works. Similar temporal logic, as surveyed by de Kleer in [deKleer, Brown 1986], has been used in the past to extend the problem solving abilities of traditional planning systems (see [Allen 1991], [Iwasaki, Simon 1986] and [deKleer 1986]).

As noted by Moray in [Moray 1990], four kinds of cause have classically distinguished (with classically meaning in the sense of going back at least to Aristotle). A switch may cause a pump to operate because it is in the "on" position *(formal cause)*, because it closes a pair of contacts *(material cause),* because it allows current to flow through the pump *(efficient cause),* or because we need cooling *(final cause).* In this thesis we are discussing an attempt to learn about *formal causality,* although extending this work to consider the other forms of causality is an intriguing prospect.

While some causality relationships can be specified a priori, many others must be learned during the creature's lifetime. To learn these relationships, the creature needs to collect statistical information about its sensory input. It needs to filter its representation of the world state and consider only the most interesting things it perceives. It then needs to discover apparent temporal causality relationships in those perceptions, and use that knowledge to inform its action selection decisions. While performing action selection, the creature needs to reinforce prediction mechanisms that are doing a good job, and refine or remove those that are unreliable.

In Section 3.3, we consider what it will take to build a creature that does these things, and as a result makes use of apparent temporal causality.

3.3.1 First of all, we represent Stimuli

A Stimulus is a signal provider in the creature's brain that can serve as a component of an apparent temporal causality relationship. The stimulus can thus represent a wide range of potential signals, from

Percepts indicating external world state, to some component of self-action, or an Autonomic Variable representing a facet of the creature's internal state.

We ask that any signal provider that backs a stimulus also provide an *activation threshold*. Much discussion in behavioral psychology revolves around the animal's perception of the "onset" and "offset" of a stimulus, suggesting that at some point, the creature distinguishes between the boolean presence or nonpresence of a stimulus. Thus, any object that will be used to back a stimulus must be capable of indicating whether or not it is currently active.

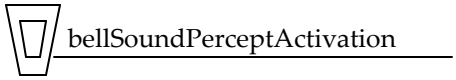


Figure 15: How stimuli are depicted in this document. The onset of the depicted stimulus occurs when the activation level of the bellSound Percept exceeds the activation threshold.

In order to discover apparent temporal causality relationships, we will need to keep some statistics about the relationships between stimuli. For example, in order to implement Rate Estimation Theory as proposed by Gallistel and Gibbon, we generate the temporal concurrence matrix, which consists of the cumulative duration of the conjunction of each pair of stimuli. In principle, this is sufficient to implement RET, as it allows us to compute the true rate of reinforcement for any stimulus using the technique described in Section 2.4.1.3. However, as we shall see later on, our implementation uses heuristics to approximate many of the underlying principles of RET.

3.3.2 Predictors represent apparent temporal causality relationships

Now that we have a means for representing stimuli, we need a way to represent apparent temporal causality relationships between those stimuli. We introduce the *Predictor,* our representation for a nugget of apparent causality information, which provides the basic unit of prediction in the system.

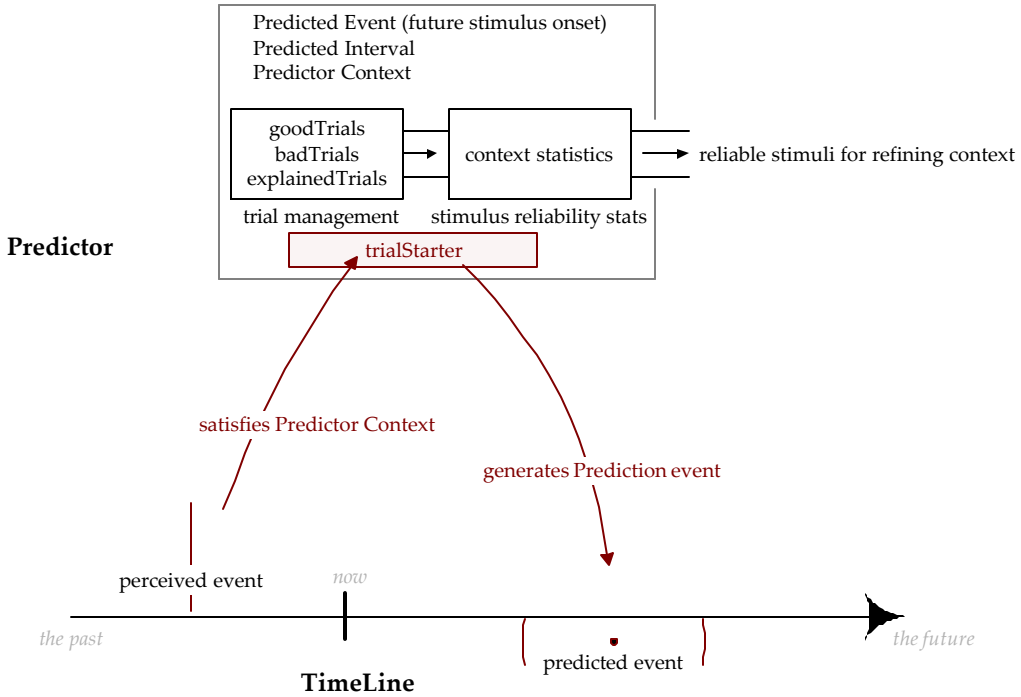


Figure 16: A Predictor interacting with the TimeLine.

A Predictor represents an apparent temporal causality relationship by *recording the perceived interval between two events*. The first event is recorded as the *Predictor Context* that consists of one or many stimuli denoting both external and internal context. The second event is recorded as the *Predicted Event* that is expected to occur in the future, after the *Predicted Interval*.

In Figure 16 we see the basic interaction between a Predictor and the TimeLine: when an event occurs that causes all of the stimuli that comprise the Predictor Context to become concurrently active, the Predictor begins a *Trial*, causing the expectation of a future event. Just as *a Predictor is analogous to SET's Reference Memory* because it stores the perceived interval between two events, *a Trial is analogous to an instance of SET's timing mechanism*, in that it represents an individual prediction that the Predicted Event will occur, with a given reliability, during a particular time window. Also directly analogous to Scalar Expectancy Theory, *the size of the time window during which the event is predicted to occur is determined simply by a scalar function of the Predicted Interval*. The importance of this property is depicted in Figure 17, where two Predictors with significantly differing Predicted Intervals produce predictions of events that are expected to occur within time windows of substantially different size.

In the Figure, the perceived event at the present time satisfies the Predictor Context for the two (unrelated) Predictors. Each of them begins a Trial, resulting in the prediction of two future events after their two Predicted Intervals. The Predicted Interval for Predictor 2 is twice the length of the Predicted Interval for Predictor 1, and so the size of the windows in which their two future events are predicted reflects this.

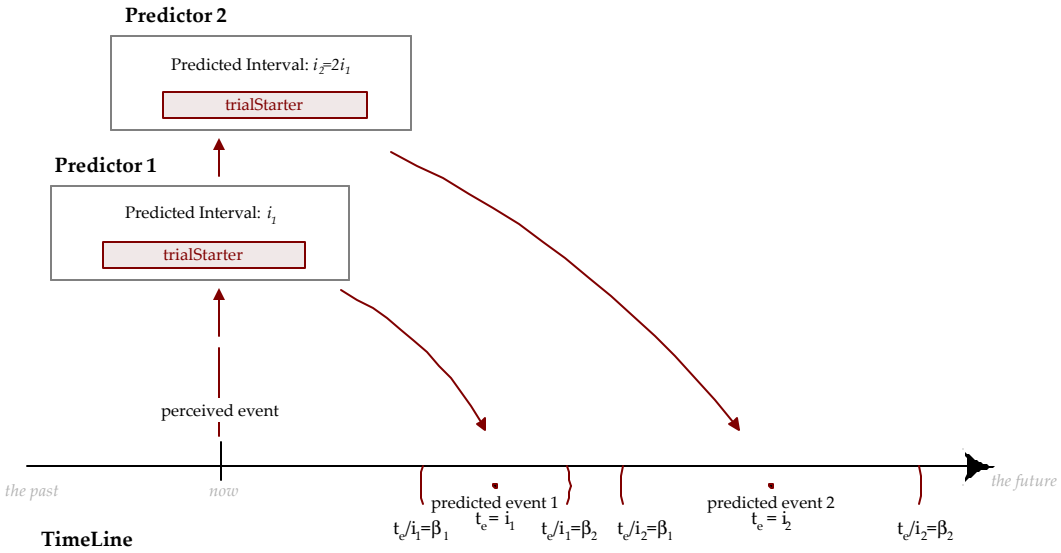


Figure 17: The timing of a Pre dictor's window.

Figure 17 also illustrates the mathematics of how the *decision thresholds*, ß1 somewhat less than 1 and ß2 somewhat greater than 1, are used for the "when decision" to decide when the subject should respond. When the ratio ($t_e/i_{predictor}$) between the subjective duration of the currently elapsing interval (t_e, which has its zero at the time when the Trial begins) and the interval encoded in the Predictor ($i_{predictor}$) exceeds the decision threshold (ß1), the creature begins to expect the appearance of the predicted stimulus. When the ratio exceeds another decision threshold (ß2), the Predictor ceases to predict the event, and generates an expectation violation response. Thus the Predictor effectively generates a "window" in which it predicts an event will occur. The window's dimensions can thus be mathematically described as:

$$b_1 i_{predictor} \le t_e \le b_2 i_{predictor} \quad (1)$$

where

$i_{predictor}$ is the *Predictor Interval in the Predictor that began this Trial*

$ß_1$ is *a creature-global constant slightly less than 1*

$ß_2$ is *a creature-global constant slightly greater than 1*

t_e is the *elapsed time since the Trial began (when the Predictor Context was met)*

Some subtlety exists in the moment at which we start a Trial. If the stimuli in the Predictor Context denote an external context – for example, hearing a bell ring – we begin a Trial when those external conditions are perceived. If they instead denote self-action – for example, the act of pulling a lever – we begin a Trial when the creature's motor system begins to perform that action. (Intent to perform the action is insufficient, as the creature may be interrupted before its motor system is able to carry out the request.) The Predictor Context can also include a combination of external and internal conditions – for example, it may require that the creature pulls a lever when a bell rings in order to start a Trial.

An ongoing Trial can expire in one of three ways. If the predicted event does occur during the time window as expected, the Trial is declared *successful*. If, without explanation, the predicted event fails to occur within the time window, the Trial can be declared a *failure*. If the predicted event fails to occur, but an external mechanism can provide an explanation for why the Trial failed, the Trial is declared *explained*. An example of an explained Trial is one in which the event fails to occur during the predicted time window, but instead appears shortly before or after that window. Instead of calling that Trial a failure, we declare it explained.

The Predictor keeps track of its short- and long-term reliability by recording the number of successful, explained and failed Trials it has generated. We'll now see how this allows Predictors, through a process of reinforcement, to learn about causality on the job.

3.3.3 How to Learn Causality on the Job

Although some Predictors might be built offline, thus representing apparent temporal causality relationships the creature knows a priori, much of this knowledge must be learned during the creature's lifetime. We now describe how a creature comes to generate and refine a new Predictor.

3.3.3.1 Concern yourself with interesting things

The first thing we need to do in order to learn about apparent temporal causality is to concern the learning mechanism with only the most interesting things.

An immediate challenge for anything but the most trivial of systems is the tremendous size of the perceptual state-space. Each stimulus might be considered another dimension of a massively multidimensional space that is probably only sparsely populated with areas of perceptual interest. We thus need a filter that separates the interesting from the uninteresting stimuli.

We use two heuristics for determining whether or not a stimulus is interesting. A stimulus can be interesting on the basis of its *novelty* (how often it has been perceived) and its *inherent salience* (as reported by its signal provider). Or, the action selection mechanism can report that it finds a stimulus interesting for other reasons. For example, a stimulus that can be temporally correlated with important consequences may be of interest, even though its inherent salience is low.

Adding a salience filter between perception and action selection provides an important barrier that lets the creature focus on the important things – the things it might find useful to learn about.

3.3.3.2 Generate Predictors to explain unexpected Stimuli

The creature would like to be able to predict changes to the stimuli it considers interesting. In fact, it is the inability to explain changes to an interesting stimulus that prompts a creature to learn. In terms of the action selection mechanism described in Section 3.2, if the react operation is unable to find a Predictor that explains a salient stimulus onset, it is provided with an opportunity to consider generating a new explanation.

Explanation generation is guided by salient events that are temporally proximate to the unexpected stimulus. Recent Perception events on the TimeLine provide a convenient collection of all such candidates. To generate the appropriate Predictor, we need simply identify the stimulus (or group of stimuli) that seems the most likely explanation for the appearance of the unexplained stimulus. That stimulus may represent some component of self-action, or some perception that has its origins in the external world.

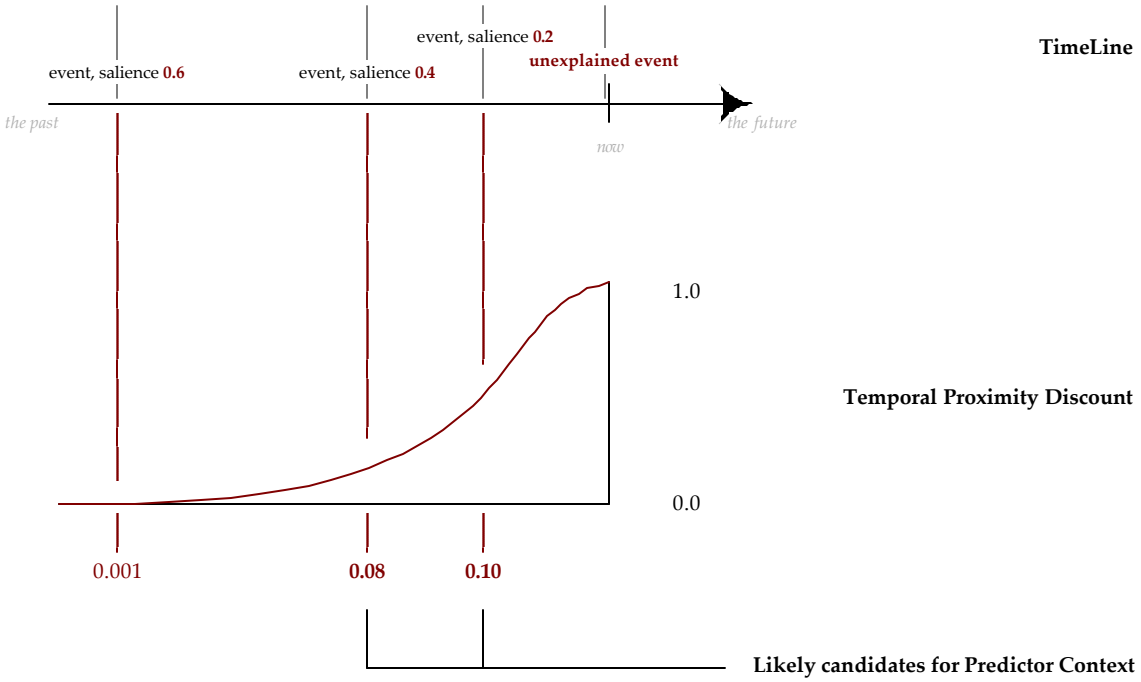


Figure 18: One possibility for selecting a likely context during Predictor generation. We model the choice of a likely explanation with the tail of a Gaussian stretching from the present into the past. We assign each of the recently-perceived stimuli a value produced by multiplying the appropriate value from the Gaussian tail with the inherent salience of the stimulus. We add these values to a probability distribution from which we select a reasonable explanation.

As illustrated in Figure 18, the explanation generator chooses a likely explanation that is both *salient* and *temporally proximate* to the unexplained stimulus. It then builds a Predictor of the unexplained stimulus with this explanation as its context. Although the particulars of function used to select an explanation are unimportant, the probabilistic nature of its operation is crucial. It is impossible in most complex systems to determine with absolute certainty which of the potential formal causes produced the perceived effect.

The length of the Predicted Interval recorded in a new Predictor is equal to the perceived length of the time between the selected explanation and the unexplained event. Thus, when creating a Predictor, we effectively initialize it with its first "Trial" – the Trial that caused its creation. It's very possible that this recorded interval is not optimal, so we allow a recorded interval to drift toward the intervals perceived

in future every time there is a successful Trial; in other words, every time it gets a new data point. The interval update equation that updates the recorded interval in a Predictor upon a successful Trial is

$$i_n = i_{n-1}(kR_{predictor}) + t^*(1 - kR_{predictor}) \quad (2)$$

where

i_n is the *new interval length*

i_{n-1} is the *previous interval length*

t^* is the *perceived interval of this Trial (see Section 2.4.1)*

$R_{predictor}$ is the *Reliability of this predictor (to be discussed; see equation* (3) *ahead)*

k is a *constant slightly less than 1.*

3.3.3.3 Refine Predictors by tracking their reliability

After we generate Predictors, we need to track their reliability. If we find that a Predictor is reliable, our confidence in its predictive power will increase. On the other hand, if the Predictor is unreliable, we may either declare it invalid, or choose to refine it in an attempt to improve its effectiveness. One method for allowing a Predictor to detect a change in a non-stationary predictive relationship would be to encode both its *recent* and *long term* reliability, so that it can detect when those reliabilities differ.

The ability to distinguish between periodicity and probability is also important. A Predictor able to ideally predict a periodic reliability schedule (for example, predicting that the event will appear on every third trial) also requires a periodic function detector (like the one described in [Aittokallio, Gyllenberg et al. 2000]). Note that a Predictor that expects an event to occur once every four times its context is observed causes quite different expectations than does a Predictor believed to be valid twenty-five percent of the time on a fixed ratio schedule. The former will generate *a high-confidence expectation every fourth time* the Predictor Context is observed; the latter will generate *a low-confidence expectation every time* the Predictor Context is observed.

A very simple metric for the long-term reliability of a Predictor is

$$R_{predictor} = \frac{g_T + e_T}{g_T + e_T + b_T} \quad (3)$$

where

g_T is the *number of successful Trials*

e_T is the *number of explained Trials*

b_T is the *number of failed Trials*

The Short-term reliability can be computed similarly by taking into account only several of the most recent Trials.

A discrepancy between the recent and long-term reliability of the Predictor might alert us to one of several possible circumstances. It is possible – perhaps because of a change in the outside world – that the Predictor has become erroneous. (This may happen frequently when dealing with a non-stationary predictive relationship.) We may conclude that this spurious Predictor should be culled. Or, we may wish to refine its Predictor Context, perhaps by a conjunction, generalization, or discrimination, in order to improve its accuracy.

We guide the refining of a Predictor by determining the *reliability* and *frequency* of salient stimuli that are observed at the onset of the Predictor's Trials. If a particular stimulus (or the onset of a particular

stimulus) is both frequently present at the onset of successful Trials (those trials where the stimulus appears as predicted), and frequently *not* present at the onset of Trials that fail, then that stimulus could potentially be added to the Predictor Context.

At the completion of a Trial, the Predictor looks back along the creature's TimeLine at a window around the Trial's start, and records all the events that occurred around that time. At the same time, the Predictor also records salient stimuli that were present in the creature's *target object*. All of these stimuli are potentially reliable indicators of the Predictor's validity: those found in the external world, in self-action, and in properties of the target of that action.

From these recorded stimuli, we need to select the ones that could be most useful if added to the Predictor Context. Candidates should be *salient* and *reliable*, meaning *frequently present* during successful trials and *frequently not present* during unsuccessful trials. We offer two metrics that satisfy those conditions.

Let g_T denote the count of successful trials, and b_T denote the count of failed trials. If g_a denotes the number of times a stimulus a was present during a successful trial, and b_a denotes the number of times a was present during a failure, then the equation

$$R_{a1} = \frac{g_a (b_T - b_a)}{(g_T b_T + 1)} \qquad (4)$$

where

g_T is the *number of successful Trials*

b_T is the *number of failed Trials*

g_a is the number of times stimulus a was present during a successful Trial

b_a is the number of times stimulus a was present during a failed Trial

satisfies these features. The first factor (g_a/g_T) provides the ratio of successful trials in which the stimulus was present; the second factor (b_T-b_a/b_T) provides the ratio of unsuccessful trials in which the stimulus was *not* present. The additive term in the denominator prevents division by zero before we have at least one good trial and one bad trial. Thus the reliability *increases* as the stimulus is present in successful trials, and *decreases* as the stimulus is present in unsuccessful trials.

Another formulation, which treats a rare stimulus more favorably, is as follows:

$$R_{a2} = \frac{1}{2(g_T + b_T)}\left[2g_a + 1[g_T + b_T - (g_a + b_a)] + 0(b_a)\right] \qquad (5)$$

In this formulation, the stimulus is effectively rewarded two points for every time it appears in a good Trial (the first term), one point every time it does not appear concurrent with a Trial (the second term), and no points when it appears concurrent with a Trial that fails (the third term). The multiplying factor scales the results to the range 0 to 1. The intuition works as follows: each stimulus gets a default of 1 point for every Trial, so every stimulus will receive the same score by default. But, if the stimulus appears concurrent with the Trial and the Trial is successful (g_a), the stimulus scores 2 points instead of 1. And if the stimulus appears concurrent with an unsuccessful Trial (b_a), it scores 0 points for that Trial. So again, the metric looks favorably on stimuli that are present during successful trials, and unfavorably on stimuli present during unsuccessful trials.

3.3.4 The representation of action reflect ideas about causality

We now have a representation for Prediction, but it will only be useful to the creature if the action selection mechanism can take advantage of apparent temporal causality knowledge.

Until now, our discussion of action selection has been rather general. At this point, we introduce a more concrete example of an action selection representation, so that we can show how it accommodates a dynamic representation of causality.

We introduce the ActionTuple, the fundamental representation of action originally proposed by Blumberg (see [Burke, Isla et al. 2001]), that we have augmented to include the previously-discussed Predictor representation.

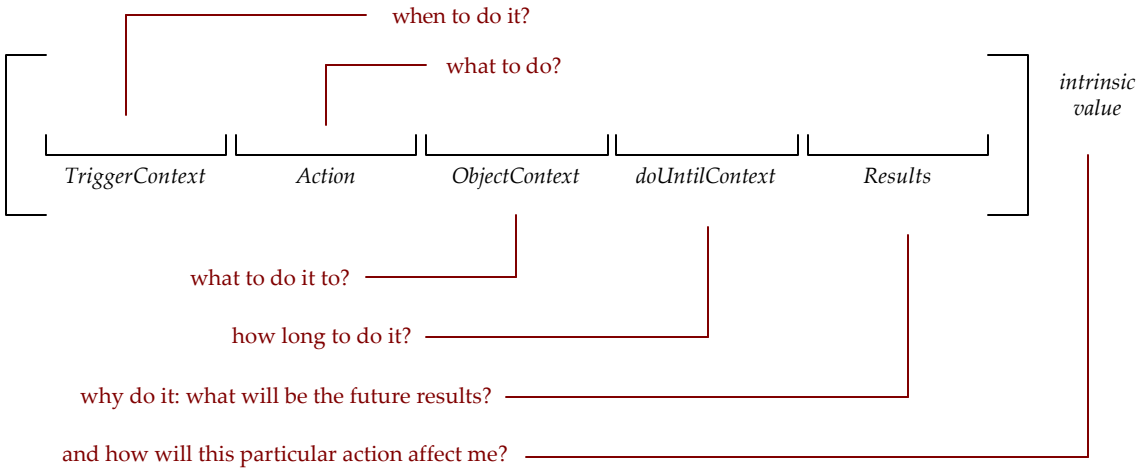


Figure 19: Anatomy of an ActionTuple. In English, from left to right by slot: "In a certain context," "if I perform an action," "on something," "for a certain amount of time," "how will it change the world," "and how will it affect my drives?"

As seen in the Figure 19, the ActionTuple encapsulates the concepts of *trigger*, *object*, *action*, *doUntil*, and in this new formulation, *results*.

The *TriggerContext* indicates external conditions that must be met in order for the ActionTuple to be activated.	*"When should I do it?"*
The *Action* represents what the creature should do if the ActionTuple is active.	*"What should I do?"*
The *ObjectContext* describes necessary conditions on the things to which that Action can be applied.	*"What should I do it to?"*
The *doUntilContext* describes the conditions that cause the ActionTuple to deactivate.	*"How long should I do it for?"*
The *Results* slot contains Predictors, as described in the previous Section, each of which predicts that when the ActionTuple is activated, an event will occur after an interval with a certain probability.	*"What will be the results?"*
The *Intrinsic Value* is a multidimensional value (with the same dimensions as the DriveVector – see Section 3.1) that describes the ActionTuple's perceived effect on the creature's Drives.	*"How will this affect my drives?"*

In Section 3.3, we indicated that each Predictor has a corresponding Predictor Context that determines when it generates expectations. We now see that the Predictors found in an ActionTuple's Results slot inherit their Predictor Context from the ActionTuple in which they are found. The TriggerContext, Action and ObjectContext slots conveniently denote external context, self-action, and the target of that action. All three of these context-denoting slots are represented by lists of stimuli. If the TriggerContext is the only one of these three slots that contains stimuli, it represents an external context, and the Predictors begin a Trial when that external context is perceived. On the other hand, if the ActionTuple contains an Action, its Predictors only begin a Trial when the creature performs that action. This prevents the creature from generating unfounded expectations if it is interrupted before its motor system has a chance to perform the requested action.

Intrinsic value is provided as a fixed value for some ActionTuples, which we refer to as *consummatory* ActionTuples. (This is a bit of a misnomer, because although high-magnitude, negative intrinsic values suggest the satisfying of drives, other ActionTuples have large, positive intrinsic values, suggesting that they will cause an *increase in the creature's drives*, thus serving as punishment rather than a reinforcer.) Consummatory ActionTuples therefore represent particular states (either action states or states defined by the context of the world) that the creature considers inherently "good" and "bad."

Although performing a particular action may not have an effect on the creature's drives, an ActionTuple's predicted Results may change the world in a way that would facilitate satisfying drives in the future. Thus the utility of an ActionTuple is defined by more than just its intrinsic value. We combine an ActionTuple's intrinsic value with the Predictors contained in its Results slot to compute its *perceived value*. The perceived value factors in the probability that the ActionTuple's activation will facilitate the future activation of consummatory ActionTuples. Importantly, a predicted Result is valuable *if and only if it will help satisfy a currently unsatisfied prerequisite* of a consummatory ActionTuple. Thus the perceived value of an ActionTuple changes as our *needs* change, *and* as the *perceived external conditions* in the world change.

The perceived value of an ActionTuple is calculated as

$$\mathrm{pv_i}(t) = \mathrm{v_i}(t) + k \sum_{m}^{predictors} \left[R_m \sum_{n}^{facilitatedTuples} \mathrm{pv_n}(t) \right] \qquad (6)$$

where

where

$\mathrm{v_i}(t)$ is the *intrinsic value of ActionTuple i*

R_m is the *reliability of each associated Predictor*

$\mathrm{pv_n}(t)$ is the *perceived value of each facilitated ActionTuple*

k is a *discount factor*

In this implementation, this perceived value equation uses a maximum recursive depth of 4. In the next subsection, we provide a concrete example of how perceived value is calculated.

3.3.5 Changing ideas about causality

We've seen how the ActionTuple combines representations of action and apparent temporal causality. We next examine how this representation can accommodate changing ideas about causality. We'll use an example to show how ActionTuples can be used to represent and learn an apparent temporal causality relationship. Consider an experiment wherein a dog is conditioned to salivate upon hearing a bell ring, because the bell provides a reliable predictor of the appearance of steak.

We begin with the assumption that the dog has the inherent idea that consuming steak will reduce his hunger drive. We construct the consummatory ActionTuple that represents this relationship (assuming the animal has only two drives, hunger and sex).

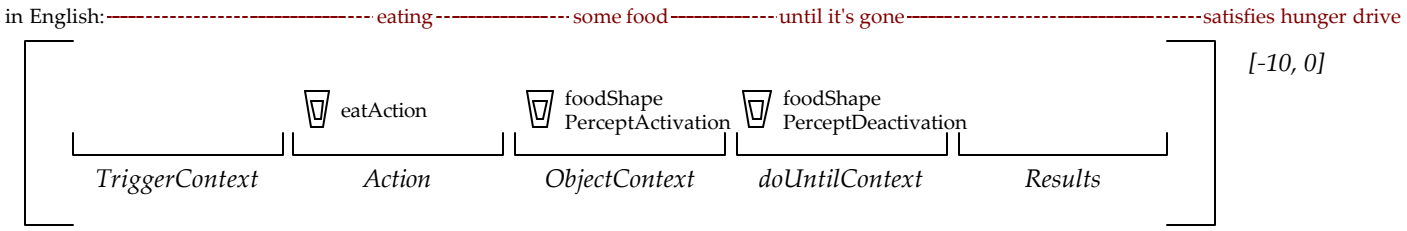


Figure 20: Consummatory ActionTuple representing eating food.

The consummatory act of eating the food is represented by the ActionTuple depicted in Figure 20: with a null TriggerContext (meaning no external conditions need to be met), the eat Action in the Action slot, the foodShape stimulus as an ObjectContext (meaning the action must be performed on food, and thus can't be performed unless food is present), and the notion "until consumed" in the doUntilContext. The intrinsic value indicates that the creature's hunger drive will be mitigated if this ActionTuple is activated. If the creature has a sufficiently high hunger drive, any sensible action selection mechanism (like the one described in Section 3.2) would be inclined to activate this ActionTuple when the creature perceives food, on the basis of its ability to reduce the hunger drive .

During this experimental procedure, the dog will be presented with two salient stimuli: the sound of the bell, and the appearance of a steak. In her attempts to explain these unexplained stimuli, the dog will, after a time, come to the idea that the sound of the bell is reliably followed by the appearance of food.

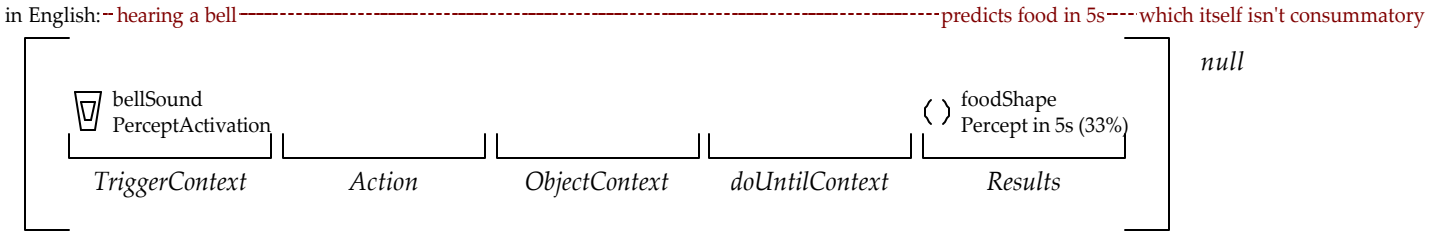


Figure 21: The ActionTuple representing the prediction that the bell sound predicts the future appearance of food.

This bell-predicts-steak notion is represented by an ActionTuple , shown in Figure 21, which produces the conditioned anticipation response when the creature hears the bell. (It turns out it that is not whimsical to speak about a real dog's "expectation of food," as evidence from Rescorla's lab, Dickinson's lab and Colwill's lab suggests that in classical conditioning protocols, subjects *do* learn to *expect the reinforcer*, using what is called a stimulus-outcome association (Gallistel, pers. comm.).) The TriggerContext for this ActionTuple is the bellSound, the ObjectContext null, the Action null, and the doUntilContext null. The Results slot contains the Predictor indicating that something with the foodShape property will appear in a few seconds , at this point with 33% reliability.

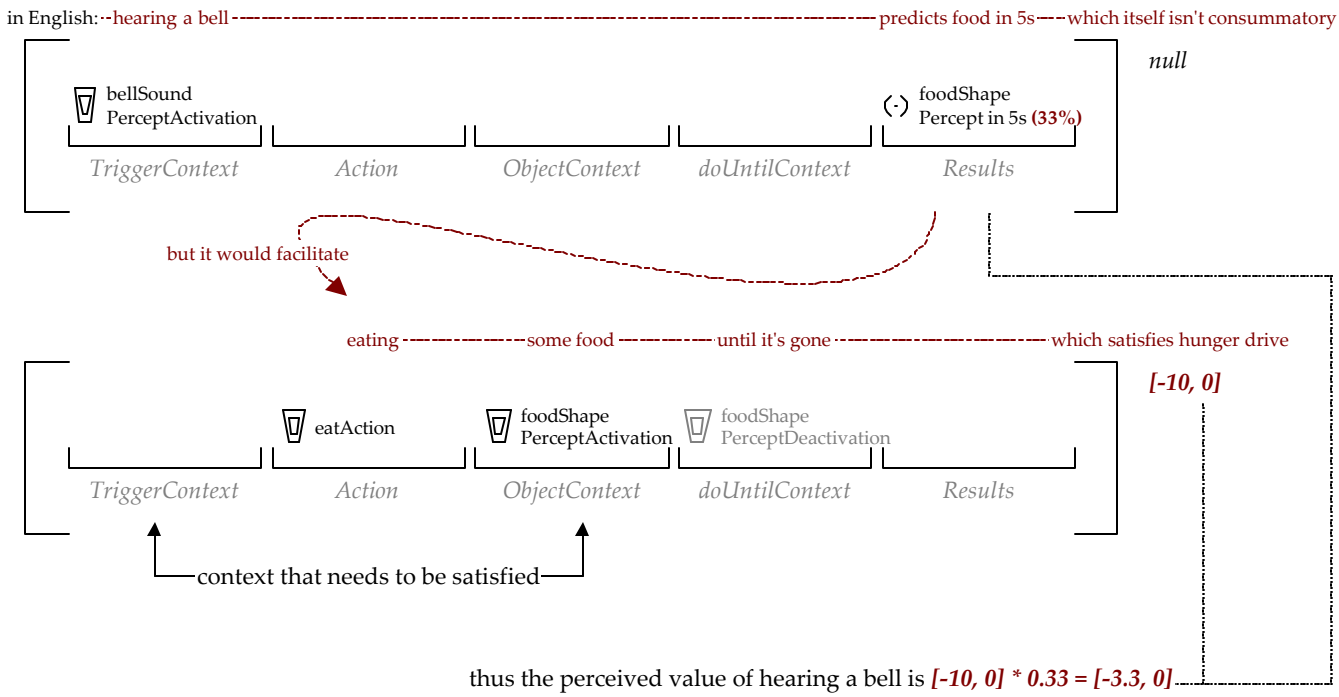


Figure 22: How the concept of perceived value makes hearing the bell a good thing.

Although the intrinsic value of the "hearing a bell" ActionTuple (depicted again at the top of Figure 22) is null (zero), the concept of *perceived value* makes its activation seem like a good thing to the dog. It indicates that the activation of this ActionTuple can reliably lead in future to the activation of another ActionTuple that will satisfy the hunger drive.

The *perceived value* of the "hearing the bell" ActionTuple is calculated by the sum of its intrinsic value, and the intrinsic values of the ActionTuples it facilitates multiplied by a discount factor. The discount factor for each term is a function of the probability that the required stimulus will appear. In this example, the dog is conditioned that food will appear on average every one in three trials. Thus the discount factor is 1/3, and since the perceived effect on the hunger drive of eating the predicted food is negative 10, the perceived value of the "hearing the bell" ActionTuple is -10/3.

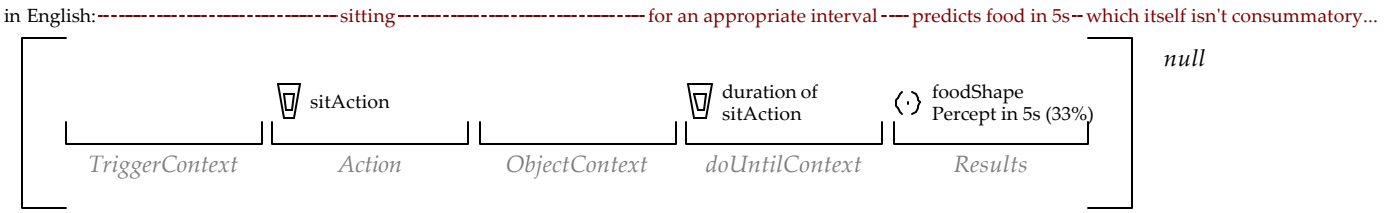


Figure 23: Self-action variation of the experiment, part 1.

A variation of this experiment might involve only providing the dog with a food reinforcer *when it sits down* after the bell rings. In this case, the dog may begin with the hypothesis (suggested by the ActionTuple in Figure 23) that simply sitting down predicts the treat. Because some of the Predictor's trials will be reinforced and others will not, the Predictor will eventually realize that whether or not the bell sounds around the start of a trial reliably predicts the trial's success.

Thus, the new ActionTuple in Figure 24 will be created. The Action is still the sitAction, but now the bell sound has been added to the TriggerContext. The Results slot, as in the above example, contains a Predictor predicting the foodShape's onset in a few seconds. Thus, *predictions can, but do not have to, involve self-action.*

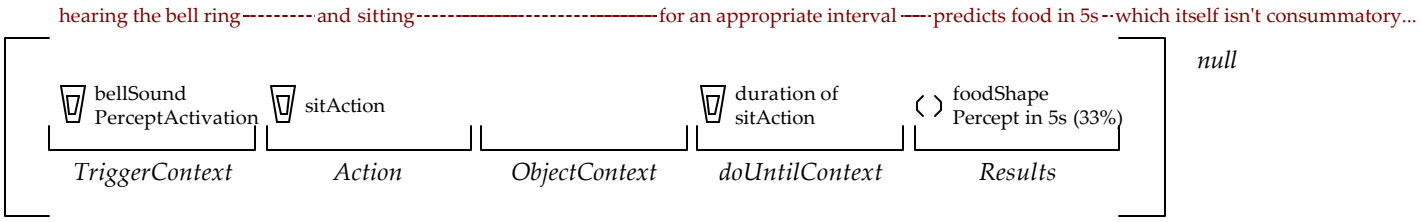


Figure 24: Self-action variation of the experiment, part 2.

3.3.6 Forgetting must accompany learning

We have discussed how the various learning mechanisms generate Predictors and ActionTuples. But as Minsky notes in Society of Mind [Minsky 1985], we have good reasons to occasionally forget things. A culling mechanism is needed to remove information that is no longer useful to the creature. In this architecture, this means removing ActionTuples and Predictors that aren't useful for predictive purposes, or have been inactive for an extremely long time.

We can think of ourselves as managing a cognitive economy. For every source (for example, of Predictors and ActionTuples) there must also be a sink, or the creature will be overwhelmed by growth. Matching each source with a sink allows a creature to adapt to changes in the world without leaving vestigial knowledge lying around.

It is exceedingly difficult to distinguish between negative and unimportant knowledge. Arguably, the most pressing avenue for future work (discussed below) is an improvement to the culling sentinel that would allow it to handle negative knowledge. Perhaps a list of "inactive" ActionTuples that were once useful would allow the creature to exhibit spontaneous recovery. The "inactive" list could continue to guide innovation without overloading the action selection mechanism.

3.4 Apparent Temporal Causality lets us model the Effects on Affect

Apparent temporal causality does more than just help us select actions. It also provides feedback from the action selection process that informs the creature's *motivational and affective state.* Not only can affective state help the creature to exhibit *intentionality* and convey its motivational and affective states to a viewer, but it also can inform the creature's action selection decisions.

Recall that a creature's DriveVector can be used to determine the scalar *utility* of something in the world. This utility can also be interpreted as a creature's *affective stance* towards something. The creature can use its *affective stance* toward a stimulus to generate appropriate reactions to its onset and predictions of its impending onset. A creature can also use the affective stance to determine whether or not it wishes to encourage the onset or offset of a stimulus. Thus an interesting effect of the DriveVector approach is that a creature's emotional memories of some context (an object, action, or whatever else) are affected by its current needs.

We have already discussed how the creature's motivational state (summarized in the DriveVector) has an effect on the action selection mechanism. Its affective state also has an influence on action selection, although not as pronounced. Affect doesn't have a direct influence on the generation, testing or refinement of Predictors. However, feedback from the Affective Variables does impact the creature's special curiosity drive. The curiosity drive, as noted in Section 3.2.1, influences the creature's decision of whether to explore or exploit, and also has an influence on the inner working of the explore

operation (Section 3.2.3). Thus, by altering the creature's propensity to explore rather than exploit, its affective state has an indirect but important effect on learning by altering the rate at which the action selection mechanism generates and refines Predictors in the explore operation.

There are four events related to apparent temporal causality that may lead to a change in affect: the perception of a salient stimulus that triggers a prediction, an expectation violation, an explanation of an expectation violation, and expectation fulfillment. Concurrent with each of these events is an appropriate change in the creature's level of *arousal*, as well as a change in *stance* proportional to the affective weight of the event.

It is obvious how the creature's level of arousal should change at each of these events. Appropriate changes to the creature's stance can be summarized as follows:

- Upon *perceiving a salient stimulus*, calculate the change in affective stance, w, based on the ability of that stimulus to predict intrinsically valuable states in the future. This generates either *eager anticipation* or *trepidation*, depending on the sign of w.
- On *expectation violations*, lose (1+k)w, where k is some number between 0 and 1. This generates *disappointment* or *relief*, depending on the sign of w.
- On *explanations of expectation violations*, gain kw back, counteracting the disappointment or relief factor.
- On *expectation fulfillment*, gain w_2-w_1, where w_2 is the affective weight of the new ActionTuple, and w_1 if the affective weight of any ActionTuple that predicted w_2. This generates *satisfaction* when the expectation comes true. (Satisfaction in the sense that the expectation was satisfied, but *not* in the sense that it satisfies the creature's drives. It is important to distinguish between effects on a creature's motivational state and its affective state – a challenge exacerbated by the fact that the vocabularies used for the two discussions often overlap!)

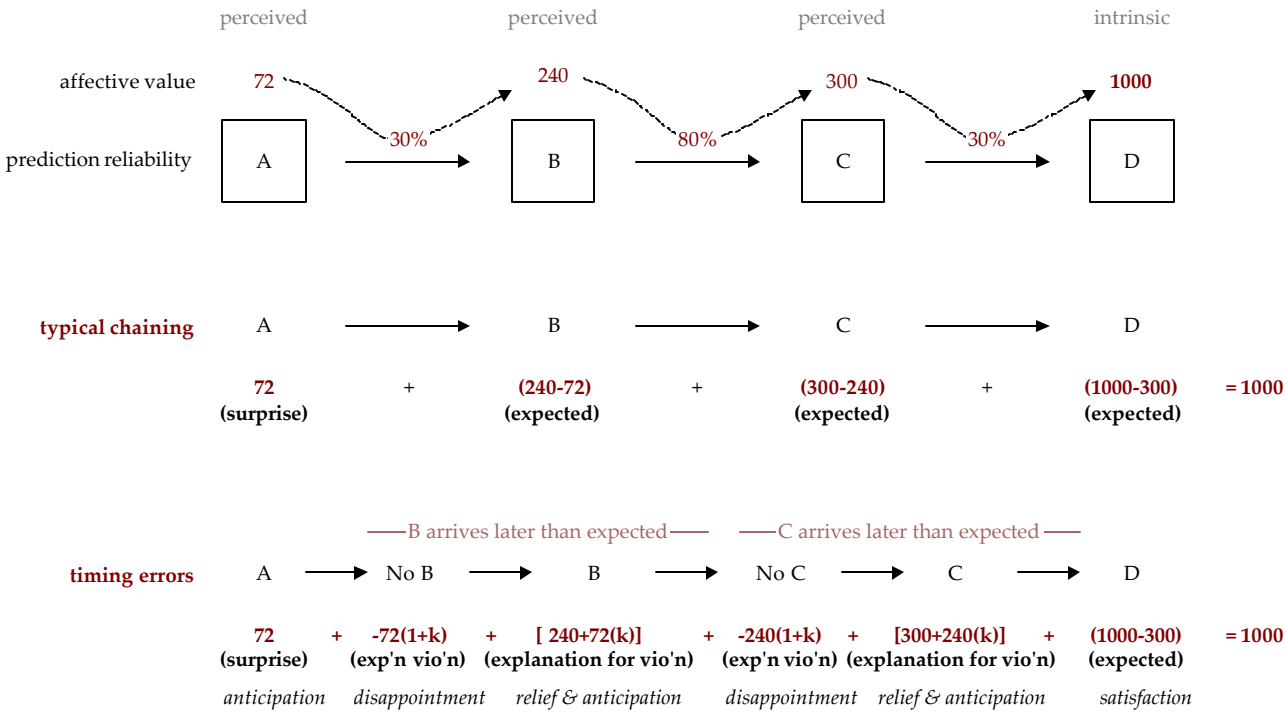


Figure 25: An example of conservation of affect during predictions, perceptions, and expectation violations.

Figure 25 provides an example of how various events cause changes to affective stance. Here we have four stimuli, A through D. A predicts B with 30% reliability, B predicts C with 80% reliability, and C predicts D with 30% reliability. D has a utility of 1000. The "typical chaining" example shows the affective changes that occur when all four stimuli appear in sequence, the former three each predicting the appearance of the next. The "timing errors" example shows what happens when the intervals the creature uses to predict B and C are too short. When each stimulus fails to appear in the predicted window, this generates an expectation violation and the creature's affect drops below the original level, thus representing disappointment. Then, the stimulus appears late, generating an explanation and thus renewed anticipation.

3.4.1 Generating New Reward Markers

Our ability to compute the affective value of a stimulus offers us flexibility in the way we produce reward markers for machine learning algorithms elsewhere in the system. Many machine learning algorithms, such as the one that drives acoustic category formation in the acoustic pattern matcher, employ a reward marker (and sometimes a punishment marker) to inform the classifier of the results of a recent classification. (See Appendix A and [Ivanov, Blumberg et al. 2000] for further discussion.)

The fundamental question is: which stimuli constitute reward markers? An obvious answer is a stimulus that indicates the appearance of a reinforcer like food. However, there may be times when we can predict the impending onset of a reinforcer with sufficiently high confidence that we can proceed to post the reward marker before the reinforcer actually appears. We do this at the moment when we can first predict, with confidence above a threshold, the future appearance of all the stimuli necessary to activate a consummatory ActionTuple. In the "ringing bell reliably predicts steakexample" depicted in Figure 22, the sound of the bell ringing may become a new reward marker.

3.5 Summary

We began this section with the notion that creatures have internal needs that they seek to satisfy. These are represented by Autonomic Variables that we combine together into a multidimensional DriveVector (Section 3.1). We then discussed the fundamental choice the action selection mechanism needs to make – whether to explore, exploit or react (Section 3.2). To help the action selection mechanism make this choice, we integrated an understanding of apparent temporal causality into the action selection mechanism. Because Predictors allow a creature to reason about apparent temporal relationships between stimuli, they allow an understanding of cause and effect that can accommodate changing ideas in a dynamic world (Section 3.3). Finally, we showed how this understanding allows us to model the effects on a creature's emotional state, and even generate new reward and punishment markers that can facilitate perceptual learning (Section 3.4).

4.0 Results for Synthetic Characters

As members of the Synthetic Characters group, we are interested in what happens when the architecture described in Section 3 is integrated into a complete system. Does it allow us to create more compelling and clever interactive creatures?

The results in this Section seem to suggest so. We describe here the results we have achieved implementing the architecture and its representations, and using them to build two very distinct autonomous virtual characters. Both characters are able to learn about apparent temporal causality in their respective worlds. Building them has provided insight into the strengths and weaknesses of our approach.

Describing these characters in more detail will provide us with an opportunity to discuss the learning process in more depth, using output obtained from visualizers that allow observers to witness the learning process occurring within a creature's brain.

All of the results in Section 4 were obtained from a working implementation. In contrast, we have yet to carry out the hypothetical experiments described in Section 5 that describe how this system should be able to reproduce psychological phenomena.

4.1 Integration into a complete Architecture

The action selection mechanism and the representations of apparent temporal causality discussed in the previous Section have been integrated into the Synthetic Characters System Architecture that we discussed in Section 2.3. (For the curious, the description in [Burke, Isla et al. 2001] is anything but brief.) In Figure 26, we show the structure of the new architecture, highlighting the important changes.

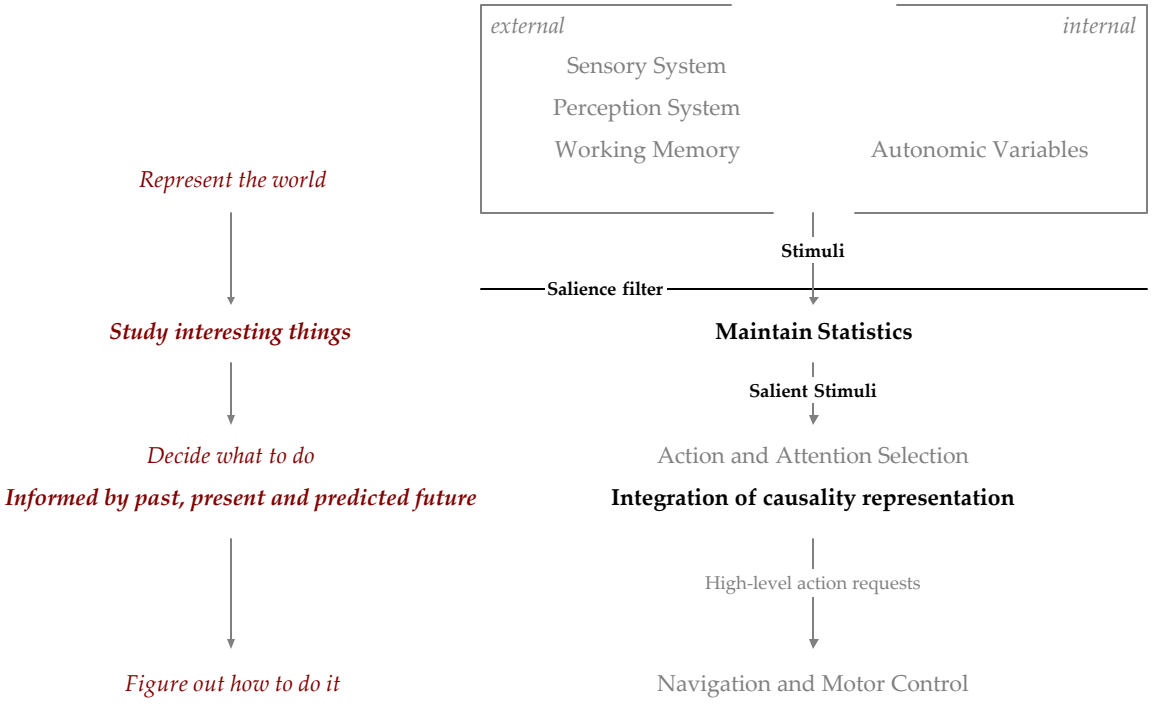


Figure 26: Integration of concepts described in Section 3 into the complete system architecture shown in Figure 2.

We discuss three aspects of the integration here that may have more general application: how the Stimulus abstraction is integrated, where we maintain statistics, and how we implement the salience filter.

4.1.1 Finding Stimuli in existing Perceptual Representations

We've already seen how the Sensory System filters the creature's sensory input, and the Perception System uses Percepts to classify and organize perceived features into an object representation in Working Memory. But how do Stimuli fit into all of this?

At first glance, it may appear that a Percept and a Stimulus are equivalent. Both have an activation threshold, and both represent an atomic component of the creature's perceptions. Suppose we wanted the creature to learn that "when the tone sounds, food will appear in five seconds." Representing this relationship involves two Percepts: the toneSound Percept, and the foodShape Percept. However, consider the relationship "when the object to my left makes a tone, food will appear." Here, the "object on my left making a tone" is represented in Working Memory as the toneSound Percept history of the "object to my left" Belief. Perhaps more importantly, the reader can no doubt come up with arbitrarily complex causal relationships that would require representations not found within this architecture. With that in mind, we propose that the Stimulus provides a useful abstraction distinct from the Percept.

We have integrated three sources of stimuli in the current system.

The first source, not surprisingly, is the Perception System's Percept Tree. Each Percept is potentially a signal provider for a Stimulus that indicates whether or not some sensory nugget has raised the Percept's evaluation above its activation threshold. Each Percept can also provide a second signal that causes a stimulus onset to occur whenever the Percept's evaluation falls *below* the activation threshold. The resulting stimulus can be used to represent relationships like "when the tone *stops*, food will appear."

The second source of stimuli is Working Memory. A Belief is capable of producing stimuli based on each of its Percept activation histories. The onset of a Stimulus based on the toneSound Percept history of the "object to my left" Belief represents the event "the object to my left made a tone."

Finally, Autonomic Variables provide a source of stimuli. They can back stimuli that allow a creature to represent apparent temporal proximity relationships involving changes to internal state.

4.1.2 A Separate Statistics and Filtering Agent

Separating the statistics from the perceptual representations proved very useful. A separate agent we call the TimeRate System provides a centralized location for the statistics kept for all the different kinds of stimuli, reducing to negligible the changes that need to be made to the other parts of the architecture in order to provide this service.

The TimeRate System treats other parts of the creature's brain, such as the Percept Tree and Working Memory, as *stimulus providers*. On each timestep, it asks stimulus providers if they have discovered any new *StimulusBackings* – signal providers that the TimeRate System can turn into Stimuli.

The TimeRate System also implements the salience filter (described in Section 3.3.3) that exists between the massively multidimensional sensory input space, and the action selection mechanism that follows it.

4.1.3 Roadmap for the rest of the Section

The best way to describe the implementation in more detail is to show a how creature implemented with the system learns about apparent temporal causality in its world. With this in mind, we now introduce the two rather dissimilar characters (at least in terms of morphology) that have been built to date using the architecture. First, we describe how the timing mechanisms facilitate new, more flexible learning in Duncan the Highland Terrier. Then, we introduce a new character, and demonstrate in depth the learning process occurring in that character's domain.

4.2 How Time Learning Affects Duncan

Duncan the Highland Terrier is one in a long line of canine creations to come from the Synthetic Characters Group.

Figure 27: Duncan the intrepid terrier.

Duncan's previous learning mechanism used a variation on the ActionTuple representation described in this paper to implement the "click and treat" training paradigm described by Wilkes in [Wilkes 1995]. With that mechanism, creatures had no sense of apparent temporal causality, but instead used *back-propagation of intrinsic value* in order to attribute value to the context-action pairs described by ActionTuples. Instead of having a concept of *perceived value* that was derived from a representation of causality, the creature would back-propagate a part of the *intrinsic value* of each activated ActionTuple to the previously activated ActionTuple. Certain consummatory ActionTuples were attributed fixed intrinsic values. These would tend to propagate their value back into the action-states that reliably led to the consummatory states. Duncan would learn by generating new ActionTuples, each time discriminating an *increasingly precise context* for the new ActionTuple which reflected which stimuli were perceived to be the most reliable indicators of whether or not the old ActionTuple's activation would lead to a "good" consummatory state.

Thus the previous incarnation of Duncan began his life by randomly selecting between different behaviors. The user, playing the role of Shep the shepherd, was able to reward Duncan's behavior and encourage him to perform particular actions, which he could eventually learn to associate with acoustic patterns by discriminating an increasingly precise context in which he should perform each action.

The representation for apparent temporal causality offers the new incarnation of Duncan many bene fits, all of which arise from his ability to make explicit how he expects the world will change as a result of his actions.

Instead of back-propagating intrinsic value, an ActionTuple obtains a perceived value because of its predictive power, which is represented by Predictors in its Results slot. The capacity for an ActionTuple to predict the appearance of food increases its perceived value if the creature is hungry. This allows the creature to integrate *the current state of the world* into its appraisal of the value of an ActionTuple. An action is only useful if it will change the world in a way that moves the creature closer to performing a consummatory action. Thus an action that facilitates the appearance of food is not valuable if there is already a vast amount of food available.

Another benefit gained from Scalar Expectancy Theory is increased flexibility in the timing of the reinforcer. Because of the nature of back-propagation learning, in Duncan's previous incarnation the reinforcer needed to immediately follow the action we wanted to reinforce. Now, if Duncan comes to expect a reinforcer to appear a slightly longer time after performing an action, he can form an expectation of future reinforcement after performing the action, and as long as the Predictor is reinforced around the expected time in the future, the perceived value of its ActionTuple would rise, and thus the correct behavior would be reinforced.

The use of a salience filter that limits the sensory input being processed by the action selection mechanism, in concert with the statistics-gathering properties of the Predictors, makes it possible for the creature to generalize as well as specify the contexts for ActionTuples. In the previous implementation, Duncan would progress down the Percept Tree, selecting increasingly specific contexts in which to perform an action, at each step taking the path that seemed the most reliable. In the current implementation, Duncan may start this process with any stimulus (that may represent any point in the Percept Tree, or perhaps some other perceptual signal), choosing a context for a Predictor that seems to be a reliable indicator of whether or not an action will lead to a reward. He can then use the results of that Predictor's Trials to generalize or specify the context from there, employing the principle of parsimony to arrive at the simplest explanation for how the world works.

Finally, while the previous incarnation would perform an action because that action was perceived to move the world into a "better state" (as represented by the intrinsic value of the active ActionTuple), creatures in this architecture make explicit their expectations of how the world should change as a result of their actions. For example, a trained Duncan will sit when the shepherd says "sit" because that action in that context will cause the explicit expectation that a food treat is forthcoming. This has ramifications on Duncan's affective model. As described in Section 3.4, expectation allows for eager anticipation (or trepidation) followed by either satisfaction or disappointment (or relief).

Thus Duncan reaps several benefits from an understanding of apparent temporal causality, all of which stem from his ability to make explicit expectations about how the world will change as a result of his actions.

4.3 The Goatzilla Domain

Goatzilla is the grizzled 200-foot-tall beast who inhabits the Scottish Highlands. What little we know of his origins is a tale passed down in the oral tradition (Eaton, Dowling, Isla, Ivanov, McDarby, McDarby, McDonnell, Nolan et. al., pers. comm.). As Duncan's master Shep tells it, Goatzilla was spawned in a freakish accident. From time to time he emerges to graze on Shep's sheep, and then stumbles off into the mist from whence he came. Shep insists that neither he nor Duncan have ever been considered targets for one of Goatzilla's feeding frenzies. Speculation abounds: one expert is adamant that Goatzilla feels guilty after eating the sheep, but a mental representation for guilt has yet to be found. Perhaps the simplest explanation is best: Goatzilla is largely misunderstood; just a creature trying to satisfy his drives. What's clear is that he can represent apparent temporal causality, and it's giving him the capacity to cause some serious damage.

Figure 28: Goatzilla and Shep share a moment.

To restore our previous tone… Goatzilla's deepest secret, of course, is that he's just a big dog. Because he provides a superset of Duncan's functionality, we use him as our primary example, and discuss his various Systems here in more detail. He exists in a complex domain where the ways in which he can satisfy his drives sometimes involve external context, sometimes involve his actions, and sometimes involve properties of the targets of his actions.

4.3.1 Perception System

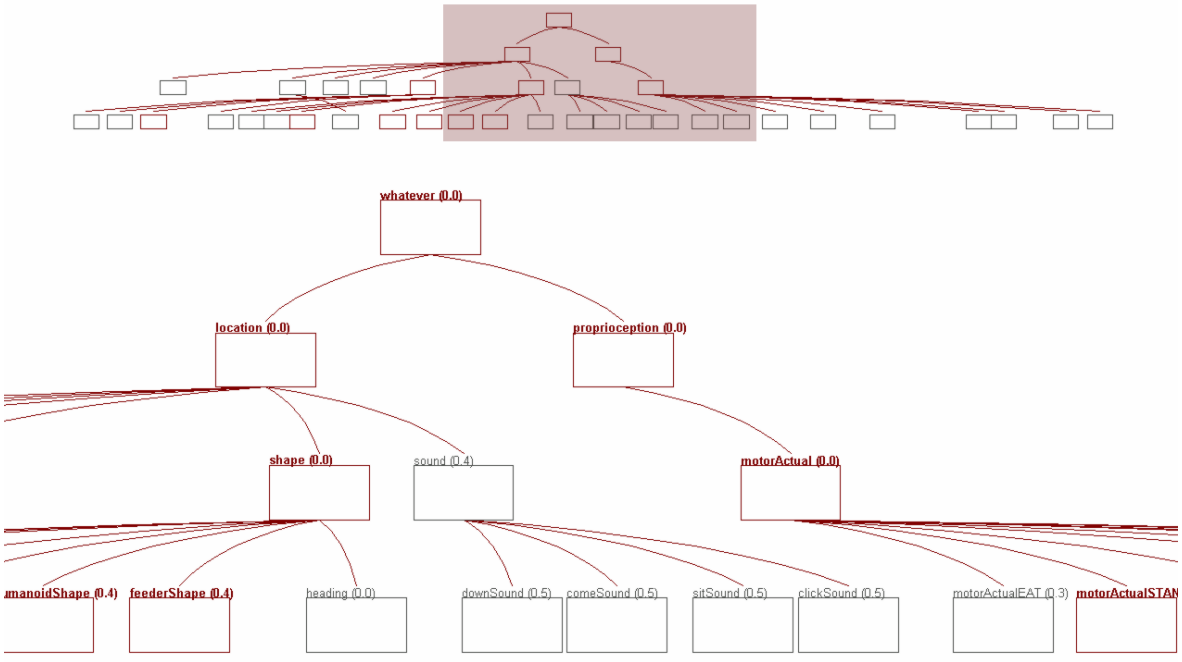


Figure 29: Goatzilla begins his life with a Percept Tree containing over 50 percepts. Colors indicate whether or not the Percept is above its activation threshold; numbers indicate inherent salience.

Highlights on the External side of his Percept tree (everything under "location" in the Figure) include two Classifier Percepts: the UtteranceClassifier Percept and the GestureClassifier Percept (neither are

visible in this Figure). Both of these are containers for classification mechanisms used to classify audio and video input respectively.

The Proprioceptive side of the Percept Tree (everything under the Proprioception Percept) contains the Percepts that activate when particular motor states are achieved. These Percepts are provided with moderately high inherent salience, so that the creature is interested in self-action and is guided to the idea that self-action is a likely candidate for apparent causality relationships.

4.3.2 Autonomic Variable System

4.3.2.1 Drives

Goatzilla has five autonomic variables that are integrated into his DriveVector. His drives provide an example of the various levels of abstraction that Autonomic Variables can represent (see Section 3.1).

Hunger is self-explanatory. It drifts upwards at a relatively low rate, has a moderate drive multiplier, and decreases when Goatzilla consumes food.

Pain avoidance has a downward drift (suggesting healing), a fairly high drive multiplier value, and increases autonomically when he is attacked, shocked, or otherwise injured. Goatzilla tends to avoid painful situations, especially if already in pain.

Dominance is a more abstract drive that provides Goatzilla's desire to dominate other creatures. The magnitude of the dominance drift increases when he perceives other Goatzillae. The drive declines when he performs various acts of dominance (posturing, kicking, goring, and so on).

Curiosity is used by the action selection mechanism to help mediate between explore and exploit operations (as described in Section 3.2).

The most recent incarnation also uses a *praise* drive that represents the social concept of being pleased to receive verbal praise from the Shepherd, which the user can provide through a special interface. Unlike the rest of the drives described here, the desire for *praise* is insatiable. The motivation for this functionality came from the need for the creature to receive praise during a sheep herding scenario, where the Shepherd wouldn't provide food treats as rewards for extended periods of time. A creature might *learn* a social drive like this by first identifying conspecifics during a critical period, and later seeking to please conspecifics perceived to be superior in a social hierarchy.

4.3.2.2 Affective Response

Both Goatzilla and Duncan use the *stance* and *arousal* autonomic variables (shown in Figure 12, page 32) to represent their current affective state.

4.3.3 Action System

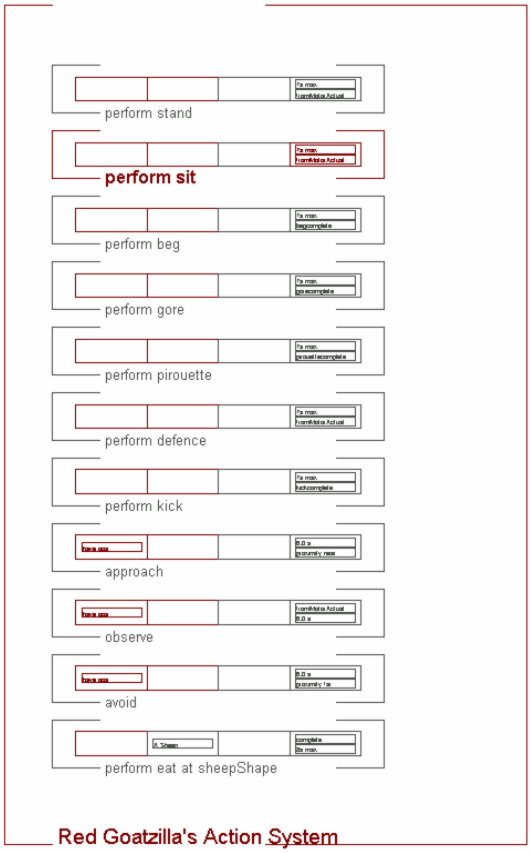


Figure 30: Visualizer of Goatzilla's default ActionTuples (without Results slots). The Reflex group is not shown.

Goatzilla begins with the simple Action System shown in Figure 30 that contains almost no a priori knowledge. Most of his knowledge is learned through experience with the world. Each of his default actions (sit, kick, pirouette, etc.) is represented by a very simple ActionTuple: a null TriggerContext, a null ObjectContext, the corresponding Action, and a doUntilContext appropriate to the Action (either one that expires at a particular interval after the motor action's activation, or one that waits for the motor action's completion).

The Approach, Observe and Avoid ActionTuples are also initialized in the Action System. The Approach ActionTuple causes Goatzilla to approach until near the target stimulus. The Observe ActionTuple only causes Goatzilla to approach the target stimulus if he is particularly far away. The Avoid ActionTuple causes Goatzilla to send a "flee" command to his Navigation System until he is far away from the target Belief.

Two examples of consummatory ActionTuples are the *"eat food"* and *"be shocked"* ActionTuples.

The *"eat food"* ActionTuple is provided with a DriveVector with a negative hunger value, suggesting that the creature has an innate sense that eating food will reduce hunger. In other words, the creature considers the activation of this ActionTuple to be the "goal state," or something that is inherently satisfying (see [Lorenz, Leyhausen 1973]).

The *"be shocked"* ActionTuple is an example of an action state that is added to the creature's group of Reflex (or "Startle") ActionTuples. Unlike the other ActionTuples that are arbitrated between using the explore, exploit and react mechanisms, when the TriggerContext of an ActionTuple in the Reflex group is active, that ActionTuple is immediately activated, and remains active until its doUntilContext is no

longer active. The ActionTuple's DriveVector contains a high, positive pain value, suggesting the creature's innate knowledge that being shocked will cause pain.

4.3.4 Navigation and Motor Control

Goatzilla's Navigation System provides the ability to override incoming "approach" and "avoid" commands with the appropriate locomotive action. His Motor System uses a verb graph representation of different animations and transitions between those animations.

4.4 Experiments: How This Works in Practice

4.4.1 Learning about the World, and Recovering from Mistakes

The first test, which demonstrates the action selection mechanism and apparent causality representations at work, involves unleashing a fresh Goatzilla into a world with the Shepherd and his Sheep. Situated beside the Shepherd are several boulders, and a shed which serves as a "feeder": when kicked by Goatzilla, it emits sheep. As the behemoth's hunger increases, he seeks and consumes any sheep he can find.

Left to his own devices in this world, Goatzilla eventually realizes that kicking the feeder reliably leads to the appearance of sheep. This is a nontrivial discovery. There are many objects in the world – sheep, the feeder, boulders, a human, a boat, and so on, each of which has the potential to generate a variety of different salient perceptual inputs. Added to which, the creature has in practice about a dozen different actions he can perform at any moment. From this considerable space of both external and internal context, the creature must arrive at the conclusion that apparently, an action (kicking) performed on a particular type of object (the feeder) is followed about a second later by the appearance of food.

We now examine in more depth how this process occurs.

When all the sheep are consumed, Goatzilla's exploration operation causes him to explore by generating new ActionTuples that perform actions on objects in the world. Those ActionTuples are generated by replicating an existing ActionTuple containing the Action of interest, and adding to the new ActionTuple an ObjectContext containing a salient stimulus from an interesting object in the world (see Section 3.2.3 on exploration). When exploring the feeder, the mechanism at some point chooses to perform the *kick* action, using the feederShape Percept's stimulus as the ObjectContext. Shortly after kicking the feeder, Goatzilla perceives the appearance of sheep.

The appearance of the sheep causes Percepts like the sheepShape Percept to be activated in the creature's Perception System. In the TimeRate System, the onset of the stimulus corresponding to the sheepShape Percept's activation occurred. For many reasons, this is a salient event: the sheepShape Percept allows the creature to activate the consummatory ActionTuple *eat food* ("perform eat at sheepshape"), an inherently good thing because of its ability to help satisfy the creature's hunger drive; and, additionally, as if that weren't enough, the sheepShape Percept's activation is unusual. Thus, the "sheepShape Percept onset" stimulus event is added to the list of salient events to be passed through the salience filter.

The onset of this "appearance of food" stimulus is passed to the action selection mechanism, where the react operation is called upon to process the stimulus onset event. The react operation polls the TimeLine for any Predictors that predict the appearance of the sheepShape Percept. Finding none, and noting that the sheepShape Percept seems highly salient, the react operation generates an explanation of the perception in the form of a Predictor. This is done by looking back at recent events on the TimeLine, and using a combination of the salience of these events and their temporal proximity to the unexpected event to come up with a potential explanation (Section 3.3.3). The creature may have come up with the right solution – that the unexplained event was a result of his previously kicking the shed, causing the results are shown in Figure 31's Predictor Visualizer.

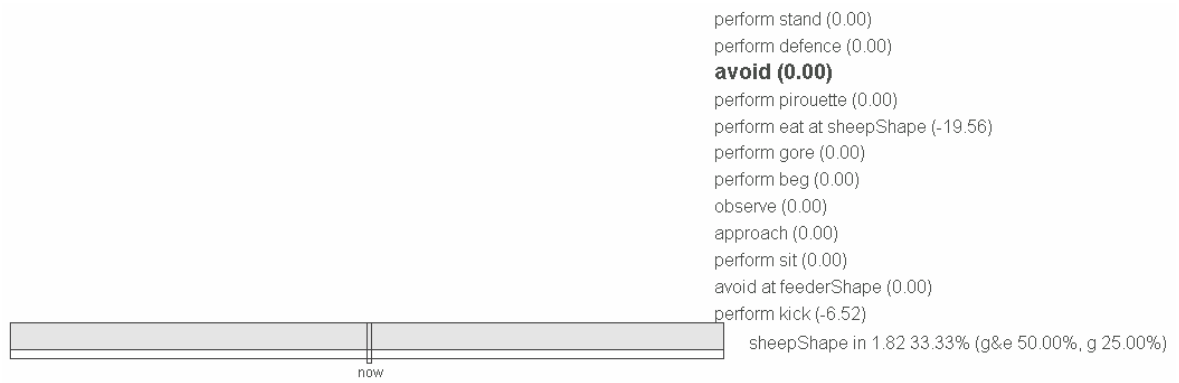


Figure 31: The Predictor visualizer. Predictors are arranged by ActionTuple, and a TimeLine for each Predictor's timing mechanism is shown. Here, the "perform kick" ActionTuple causes the prediction of something sheep shaped to appear after an interval of about 1.8 seconds with a long term reliability of 33%. Recently, 50% of that Predictor's Trials have been either successful or explained, and 25% have been successful.

We now examine how the creature can recover from any one of a number of possible mistakes.

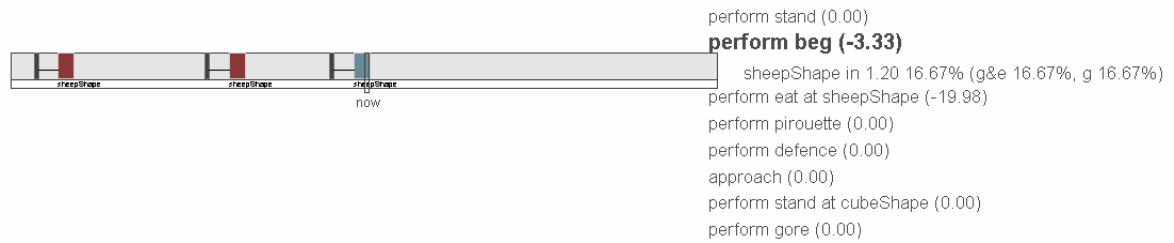


Figure 32: A Predictor that will soon prove unreliable : that begging produces sheep. Grey vertical bars on the TimeLine indicate the start of a Trial; the colored region that follows indicates the window in which the onset of the stimulus is expected to occur. Color indicates the status of the trial: ongoing blue; successful green; failed red; explained orange.

Let us assume for a moment that the creature formed *an erroneous Predictor based on another self-action*; for example, the notion that begging in front of the shed results in the appearance of the sheep. The creature would then proceed to test this Predictor, in the process producing superstitious behavior, until the Predictor proved sufficiently unreliable. When the creature is next hungry and there are no sheep present, the *"perform beg"* ActionTuple will have a high perceived value because of its perceived ability to predict the appearance of sheep a few seconds after the creature performs the begging action. The creature will activate this ActionTuple, sitting down and thus causing an expectation that sheep will appear momentarily, complete with a corresponding change in affect (anticipation). When the sheep do not appear, there will be an expectation violation and another corresponding change in affect (disappointment). As repeated occurrences of this expectation violation occur, the reliability of the Predictor will decline toward zero, until it is finally culled.

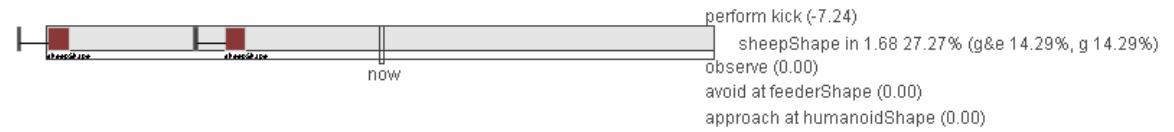


Figure 33: A Predictor with the right Action (kick), but no ObjectContext (so he'll kick *anything* and expect food).

The creature may also form a hypothesis *with the correct Action but incorrect ObjectContext* ; for example, the hypothesis that kicking *anything* leads to the appearance of sheep (an ActionTuple with the kick Action, no ObjectContext, and a corresponding Predictor in the results slot). In this case, assuming the

creature occasionally kicks both the shed and other objects in its world, the reliability of the Predictor will be reasonably high, but still result in expectation violations every time the creature kicks another type of object. After enough such expectation violations, the Predictor will almost certainly be refined by the inclusion of the feederShape Percept as an ObjectContext, creating a reliable predictor of successful trials. (Ideally, the TriggerContext will also be modified to include the qualification that the sheepShape Percept be inactive, so that the entire Predictor reads, "When no sheep are present, and I kick a feeder-shaped object, I predict something sheep-shaped will appear in *n* seconds.")

Finally, the creature may form *a correct Predictor with an incorrectly recorded interval length.* To recover from this mistake, instead of creating a new Predictor, we allow the peak of the interval recorded in the existing Predictor to drift toward the interval perceived by the creature in subsequent trials. The time scale invariance of Scalar Expectancy Theory is convenient here, as it will cause the size of the prediction window to automatically increase as the estimated interval increases.

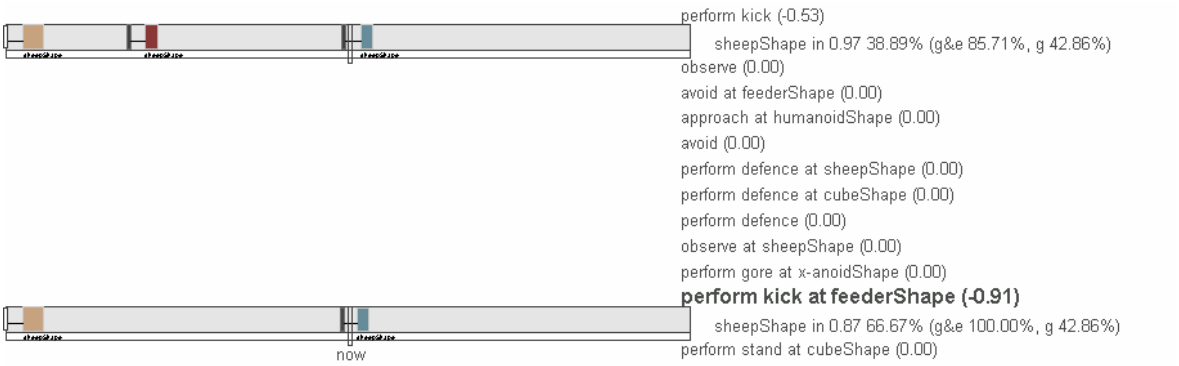


Figure 34: Learning the right thing. The "perform kick produces sheepShape" Predictor at the top of the Figure has a lower long term reliability than does the "perform kick *at feederShape* produces sheepShape" at the bottom of the Figure. After a few more Trials, the former Predictor will be culled. The active prediction suggests the creature just kicked the feeder. A few moments ago, he kicked something other than the feeder, activating the upper predictor and causing a failed trial.

To summarize, we have looked at how the creature can recover from several mistakes it might make while trying to learn a nugget of apparent temporal causality – forming an erroneous Predictor based on another self-action, forming a Predictor without a discriminating ObjectContext, or incorrectly recording the interval between stimuli.

4.4.2 Learning Curves for "Kicking Produces Food" Predictor

We examine the learning curve for a Predictor by detailing the response of the "perform kick produces sheepShape" Predictor described in Section 4.4.1 as it responds over the course of 20 Trials, each trial representing the act of kicking an object. Eventually, the Predictor refines its context to generate the more accurate and precise "perform kick at feederShape produces sheepShape" Predictor.

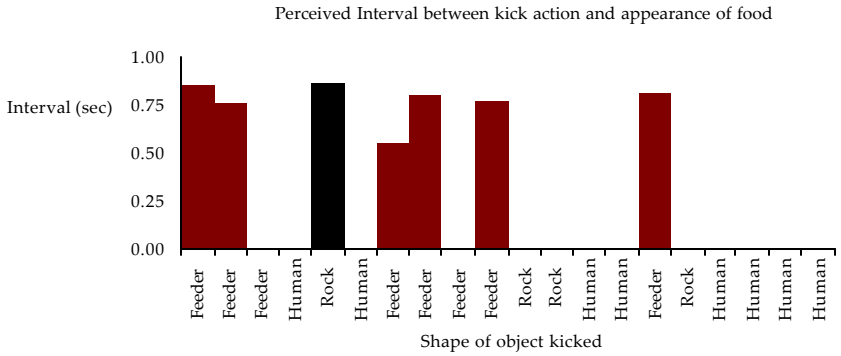


Figure 35: The results of 20 trials. Trials without bars were not reinforced, and thus declared "failures."

In Figure 35 we see the creature's perceived results of the 20 trials. The Predictor was generated at the end of the first trial, after the feeder was kicked the first time and food appeared unexpectedly. On most of the trials where the creature kicked the feeder, food appeared in approximately three-quarters of a second. On the fifth trial, the Predictor drew the erroneous conclusion that kicking a boulder resulted in the appearance of sheep. Those events, while temporally proximate, were not causally related.

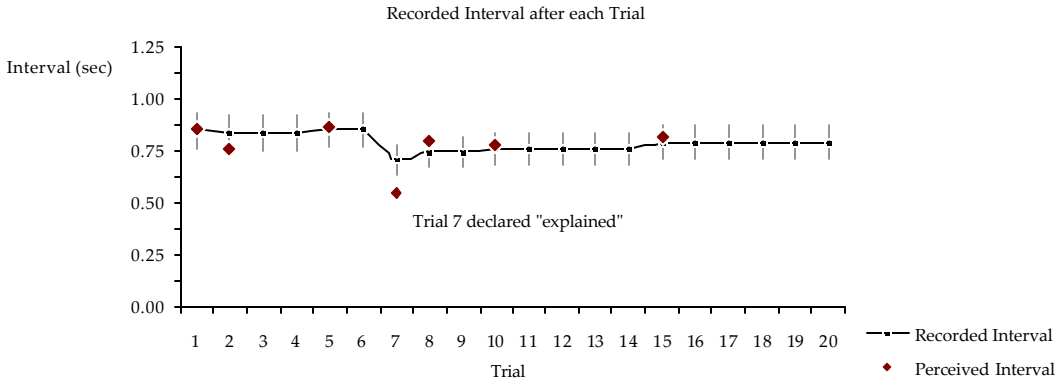


Figure 36: Learning the predicted interval.

In Figure 36, we see how the Predictor's recorded interval adjusts toward the perceived interval of each reinforced trial. The magnitude of the "drift" experienced by the recorded interval toward each perceived interval is proportional to the long-term reliability of the Predictor, a metric we show below in Figure 37.

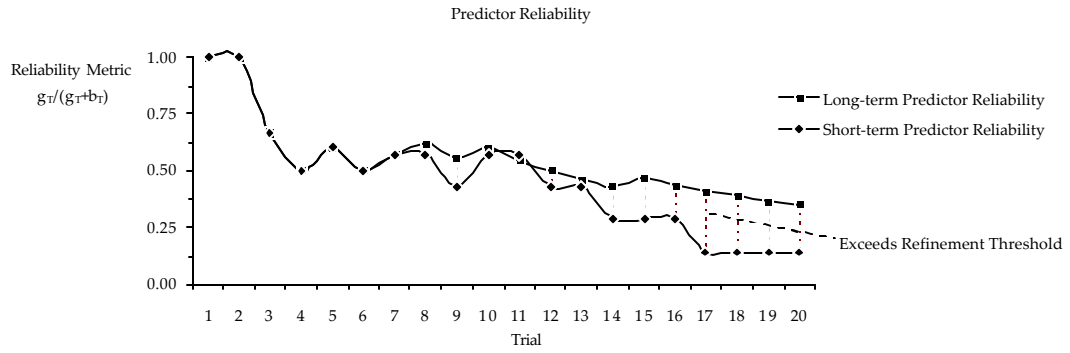


Figure 37: Long-term and Short-term reliabilities of the Predictor.

The Predictor's reliability, computed using the simple metric shown in the Figure, changes as trials succeed or fail. For the short-term metric, only several (in this case seven) recent Trials are used to calculate the reliability. At the seventeenth trial in this example, the difference between the short-term and long-term reliabilities exceeded the refinement threshold, and the Predictor was prompted to refine its context. It did so, using data like those shown in Figure 38 that indicate the reliability of stimuli that are present around the start of trials.

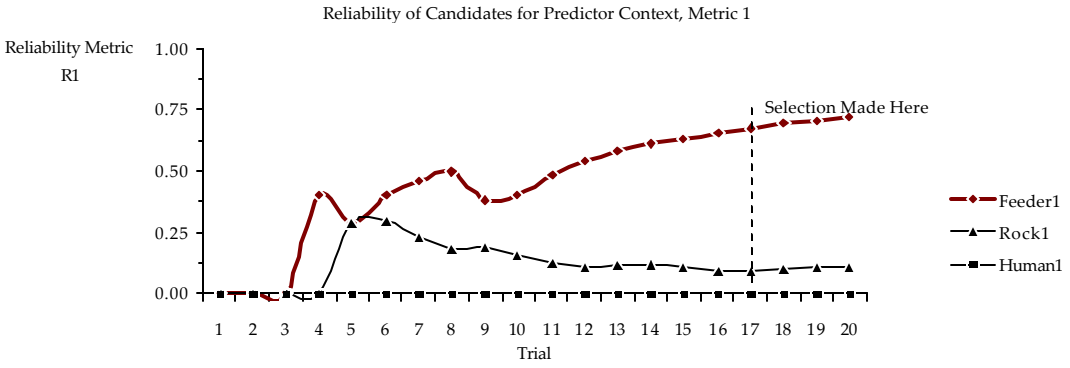


Figure 38: Reliability data for Predictor Context candidates, using reliability metric 1 (equation (4)).

For clarity, we show in this Figure only three of the salient stimuli that are potential candidates for the Predictor Context. However, it is important to note that there are many other such candidates; in fact, the Predictor is tracking the reliability of every salient stimulus. It is obvious from the data that the feeder is the most reliable candidate, and, as mentioned above, it is selected at the seventeenth trial to be added to the Predictor Context.

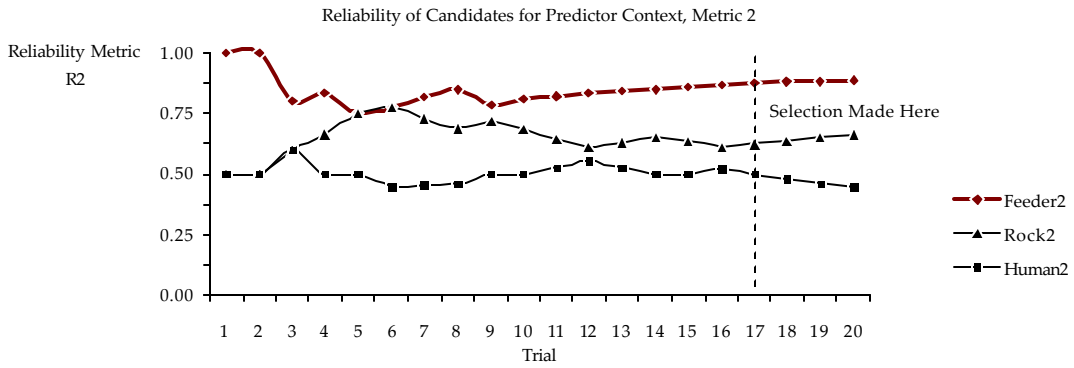


Figure 39: For comparison to Figure 38, reliability data for Predictor Context candidates, using reliability metric 2 (equation (5)).

4.4.3 Clicker Training

The time and rate learning mechanisms also provide a robust implementation of a form of reinforcement learning known as clicker training. Clicker training is form of applied operant conditioning used to train live animals like dogs and dolphins, as described by Wilkes in [Wilkes 1995].

The trainer employs a hand-held "clicker" device that emits a short, sharp clicking noise. During the first phase of training, the trainer associates the click with a food treat by clicking, and then immediately giving the creature a treat.

When the animal has learned that click precedes treat, the second phase of training begins, in which the trainer provides a click-and-treat only when the animal performs the desired behavior. The click acts as a salient event marker, indicating to the animal the precise time at which it performed the desired action. In accordance with Thorndike's Law of Effect, described in [Thorndike 1911], the frequency of the behavior that leads to the reward will tend to increase. In practice, the desired behavior does not simply appear, but rather is "shaped" by the rewarding of increasingly accurate versions of the behavior.

If some other signal, such as an utterance, should trigger the appearance of the behavior, we enter a third phase of training, in which the desired trigger signal is made to appear some of the time when the animal performs the action, and the animal rewarded only when it performs the action with the signal present. The signal then becomes a reliable indicator of the context in which the action should be performed in order for the creature to receive a reward.

The architecture described in the previous section allows for a robust form of clicker training to take place in synthetic characters. The creature is able to recover from any mistakes or incorrect guesses it makes along the way without "bogging down" its mind with useless, superfluous, or out-of-date information. We now describe how the architecture facilitates the three phases of clicker training.

4.4.3.1 First Phase: Associate Click with Treat

In the first phase of clicker training, the trainer clicks, and then follows that click with a treat. Because the TimeRate System interprets the sound of the clicker as both an unusual and salient stimulus, the clicker stimulus onset will be passed on to the action selection mechanism, causing a react operation to be performed. The creature will not be able to explain the click, and, assuming that the trainer is randomly clicking, any Predictors that are generated to explain what causes a click will eventually be discarded as unreliable .

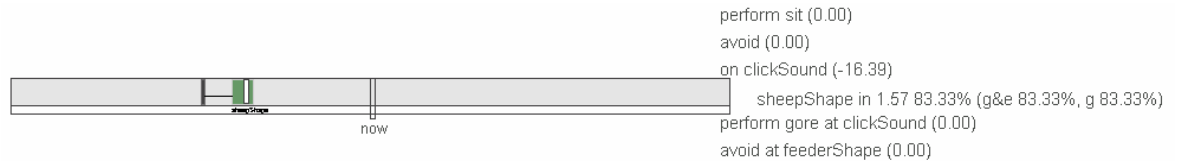


Figure 40: Results of phase 1: A Predictor indicating that the click precedes treat.

Similarly, the appearance of food will be passed as a salient event to the action selection mechanism. The creature will again attempt to explain this stimulus onset. There is a very high probability that it will form a Predictor indicating that the click stimulus precedes the food stimulus, as the onset of the click stimulus was not only highly salient, but also temporally proximate to the treat stimulus. When the creature creates the appropriate "click precedes food" Predictor, it will encode the predicted interval between the two stimuli as equal to its perception of the initial interval between those two events. During subsequent trials, the prediction interval will drift toward the average click-to-treat interval used by the trainer.

Although when training live animals with the clicker-training procedure a click should always be followed by a treat, the computational architecture will still work if this is not always the case. The creature may instead grow accustomed to a long-term reliability for the clicker much less than 100%. In fact, it is *a change in the rate of reinforcement* that motivates learning. When the creature detects a difference between the short- and long-term reliability of a Predictor, an affective change is produced that causes an increase in curiosity, and the Predictor is provided with an opportunity to refine itself.

4.4.3.2 Second Phase: Associate Action with Click

In the second phase of clicker training, the trainer clicks when the animal performs a certain action. At this time, the creature still doesn't have a reliable Predictor for the appearance of the click stimulus, but the action selection mechanism's react operation is particularly prone to finding such a Predictor, as the creature has a strong affective stance toward the click stimulus (because of its ability to predict the appearance of food).

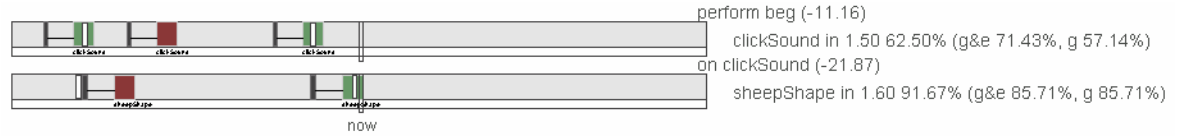


Figure 41: Results of phase 2: A Predictor indicating that begging predicts a click (top of Figure).

The creature will generate a Predictor suggesting that when it performs a particular action – for example, sitting down – an onset of the click stimulus will occur after an interval. When the creature performs the action again, the Predictor places on the creature's TimeLine a Prediction Event indicating that the click stimulus will appear momentarily. As Trials like this are successful and the Predictor is reinforced, the creature will exhibit Thorndike's Law of Effect, increasing the percentage of the time it sits down when it is hungry.

4.4.3.3 Third Phase: Associate Signal-plus-Action with Click

In the third phase of clicker training, the trainer will take advantage of the creature's propensity to perform a particular action in order to associate a particular signal with that action. The trainer causes the onset of the signal around the time that the creature performs the action, and only provides reinforcement when the creature pe rforms the action while the signal is present.

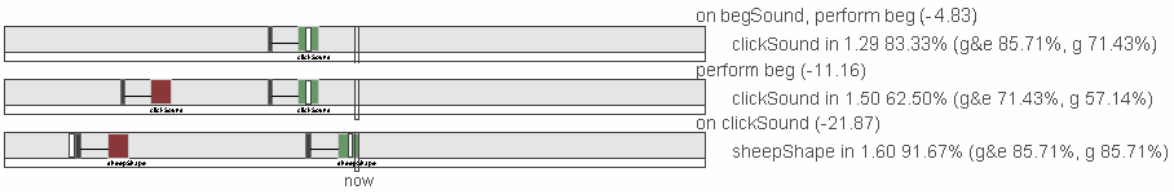


Figure 42: Phase 3: the "begging when you hear the 'beg sound' results in a click" Predictor (top of Figure) becomes more reliable than the simpler "begging results in a click" Pre dictor (middle of Figure).

The creature finds that the reliability of the Predictor it generated in the second phase of training continues to decline until its recent reliability is substantially different from its long term reliability. This will trigger the Predictor to refine its context in an attempt to become more reliable. It should note that the presence of the trainer's signal at the onset of Trials is a particularly reliable indicator of the success or failure of those Trials. Thus, the refined Predictor will include the trainer's signal in the TriggerContext.

Over time, the simpler "beg produces click" Predictor formed during the second phase of training will be culled, and the new "beg when I observe the signal produces click" Predictor will prove to be much more reliable.

4.5 Summary

In discussing how we have integrated apparent temporal causality into the architecture, we have introduced two characters – one in an applied operant conditioning domain, and the other in the more whimsical Goatzilla domain – that demonstrate in more depth how the learning process occurs. Duncan's example showed how an understanding of apparent temporal causality allows the system to perform new and more robust forms of learning. Goatzilla showed how the action selection mechanism and causality relationships allow a creature living in a complex domain to learn about its world. Both examples illustrate how the system is able to create virtual creatures that operate in real-time in non-trivial domains, solving problems, even capable of being trained like their real-world counterparts.

5.0 Results for Cognitive Psychology

In the previous Section, we described results obtained by integrating the representations for apparent temporal causality and action into an existing behavior system, and observing two characters built with the augmented architecture. We now discuss this architecture's ability to reproduce a variety of conditioning phenomena well-known to the cognitive psychology community.

We begin with a discussion of how useful and elegant Scalar Expectancy Theory's concept of time scale invariance has been, and the challenges we have faced integrating Rate Estimation Theory into a computational system. Inspired by Gallistel and Gibbon's contrast between the associative (Rescorla-Wagner) framework and the timing framework, we then provide ActionTuple answers to some basic questions from an introductory course on learning that demonstrate how our architecture differs from the associative and timing models.

Finally, we discuss this architecture's theoretical capacity to recreate cue competition and background conditioning phenomena. Although we have not yet had the opportunity to set up each of these experiments and observe the resulting behavior, we offer here a theory of how the system should perform in a variety of experimental setups.

5.1 The Utility of Time Scale Invariance

The time scale invariance suggested by Scalar Expectancy Theory provides an elegant representation of internal timing in the Predictor representation. Because Predictors record the perceived interval between two stimuli and use ß-thresholds to generate windows in which stimuli are predicted, the creature is able to predict short intervals with high precision, and exhibit a plausible amount of uncertainty for longer intervals.

Consider the alternatives. We could have used an associative framework *that did not include a temporal window*. We have already discussed (as do Gallistel and Gibbon) how this prevents us from explaining many observable conditioning phenomena. We could have used a *default interval length*, assuming that any predicted events would occur "very shortly" in the future. This technique, used with some success in the previous architecture, prevents the creature from reacting appropriately when the interval between events is more than a few seconds in length. If a shock is predicted to arrive in roughly twenty seconds, the creature should probably wait nearly that long before taking action to avoid the shock. We could have used a *variable interval length and a fixed window size*. The window size then becomes a free parameter that would have to be arbitrarily set.

A final alternative would be to derive an algorithm for simultaneously learning *both the interval length and window size*. This would allow us to express some temporal intervals that might be useful, say, for having a creature learn a piece of classical music in the western tradition, for which it would need to represent intervals of varying length with high prevision. To do this would require maintaining additional statistics every time a stimulus is perceived. Without constraints, this would produce the biologically-implausible result that creatures could learn to predict high-resolution events in the distant future. There may yet be some heuristic that would allow us to learn interval length and window size in a reasonable, biologically-plausible way. But for now, time scale invariance has helped us create an intuitive and elegant representation for the causality relationships the system has needed to represent.

5.2 The Rate Estimation Challenge

As a computational model, Rate Estimation Theory is capable of explaining a wide variety of observable conditioning phenomena in a particularly elegant way. Its underlying principles – the principle of parsimony and the principle of rate additivity – have provided the foundation of this architecture's design.

Implementing a pure form of RET's computational process (described in Section 2.4.1.3) into the computational architecture has posed two significant difficulties.

The first is one of scale: RET requires that we maintain a temporal coefficient matrix that contains stimulus concurrence information for all perceived stimuli. For a system that must contend with hundreds or possibly thousands of stimuli, the n^2 scaling factor is a challenge. There are two beacons of hope. First, this matrix is undoubtedly sparse, and it contains large regions that could possibly be approximated. Second, the salience filter's ability to reduce the size of the input space offers us an opportunity to reduce the magnitude of n.

The second challenge is one of computational complexity. Arriving at the corrected rates of reinforcement for the stimuli involves at least one matrix inversion and a matrix multiply, plus a recursion on this process for each redundant stimulus that needs to be removed. To add to this complexity, in this architecture the values of stimuli are computed relative to their perceived effect on the creature's *current* drive state, thus rendering it very difficult to cache the corrected rates of reinforcement for various stimuli, or compute those rates of reinforcement "offline" with a process happening in the background.

We have not been able to address these challenges, but our implementation provides an approximation to RET that incorporates both of its fundamental principles in the way a creature attempts to explain its world by building reliable Predictors. When trying to explain a surprising stimulus (during the *react* operation), and when new stimuli are being chosen (while refining a Predictor), rate additivity is used to assess the perceived value of a stimulus. The principle of parsimony is also fundamental in that the creature is only motivated to learn the simplest explanation for how its world works. The stimuli chosen to be added to the context of a Predictor during innovation are those that most simply and reliably allow the creature to predict the future onset of a stimulus.

One thing worth noting here is that this architecture's Predictor implementation makes it very difficult for the creature to represent random rate processes, something we discuss in more detail under "Future Work" in Section 7.3.1.

The best way to illustrate how the approximation heuristics work is to take a cue from Gallistel and Gibbon, and compare and contrast this architecture with the associative and timing models.

5.3 Different Answers to Basic Questions Redux

Inspired by Gallistel and Gibbon's contrast between the associative (Rescorla-Wagner) framework and the timing framework, we provide ActionTuple answers to some basic questions from an introductory course on learning. The standard and timing answers are reproduced from [Gallistel, Gibbon 2000].

1. Why does the conditioned response appear during conditioning?

Associative answer: Because the associative connection gets stronger.

Timing answer: Because the decision ratio for the whether-to-respond decision grows until it exceeds a decision threshold.

ActionTuple answer: Because the confidence in a Predictor is sufficiently high to trigger a response during a react operation.

2. Why does the CR disappear during extinction?

Associative answer: Because there is a loss of net excitatory associative strength. This loss occurs either because the excitatory association itself has been weakened or because a countervailing inhibitory association has been strengthened.

Timing answer: Because the decision ratio for the whether-to-stop decision grows until it exceeds the decision threshold.

ActionTuple answer: Because the confidence in a Predictor declines until it is below the culling threshold.

(The extinction protocol is one wherein *the CS is presented without the US until the CR disappears.)*

3. What is the effect of reinforcement?

Associative answer: It strengthens excitatory associations.

Timing answer: It marks the beginning and/or termination of one or more intervals: an inter-reinforcement interval, a CS-US interval, or both.

ActionTuple answer: It marks the beginning and/or the termination of one or more intervals: an inter-reinforcement interval, a CS-US interval, or both.

4. What is the effect of delay of reinforcement?

Associative answer: It reduces the increment in associative strength produced by a reinforcement.

Timing answer: It lengthens the remembered inter-reinforcement interval, the remembered CS-US interval, or both.

ActionTuple answer: It lengthens the remembered inter-reinforcement interval, the remembered CS-US interval, or both.

5. What is the effect of non-reinforcement?

Associative answer: The non-reinforcement (the No-US) weakens the excitatory association; or, it strengthens an inhibitory association.

Timing answer: The timer for the most recent inter-reinforcement interval continues to accumulate.

ActionTuple answer: The timer for the most recent inter-reinforcement interval, and the timers for any Predictions on the TimeLine predicting the appearance of reinforcement (each corresponding to a Predictor Trial), continue to accumulate. When significantly past the predicted time of a reinforcement, the Trial associated with a Prediction will be flagged as a failure, reducing the corresponding Predictor's predictive confidence.

6. What happens when nothing happens (during the inter-trial interval)?

Associative answer: Nothing.

Timing answer: The timer for the background continues to accumulate.

ActionTuple answer: Any timers from any TimeLine Predictions continue to accumulate, as do the creature's background timer (its internal clock).

7. What is the effect of CS onset?

Associative answer: It opens the associative window in the mechanism that responds to the temporal pairing of two signals. That is, it begins a trial during which the updating of associative strengths will occur.

Timing answer: It starts a timer (to time the duration of this presentation) and it causes the cumulative exposure timers to resume cumulating.

ActionTuple answer: Each Predictor with the CS as its Predictor Context starts a Trial, causing a Prediction Event to be placed on the TimeLine and its timer started. The cumulative exposure timers for each Stimulus (and any concurrence duration exposure timers) resume cumulating, although the cumulative exposure times are not used in this architecture.

8. What is the effect of varying the magnitude of reinforcement?

Associative answer: It varies the size of the increment in the excitatory association.

Timing answer: It varies the remembered magnitude of reinforcement.

ActionTuple answer: Predictors predicting various magnitudes of reinforcement are generated. These magnitudes are assumed to be discretized by the available Percepts in the creature's Percept Tree (big reinforcement, small reinforcement, etc.) The expectation violation results in an increase in the curiosity drive, encouraging exploration.

9. Why is the latency of the conditioned response proportional to the latency of reinforcement?

Associative answer: There is no widely accepted answer to this question in associative theory.

Timing answer: Because the animal remembers the reinforcement latency and compares a currently elapsing interval to that remembered interval.

ActionTuple answer: Because the animal remembers the reinforcement latency and compares a currently elapsing interval (that of the Prediction found on the TimeLine) to the remembered interval (the interval encoded in the corresponding Predictor).

10. What happens when more than one CS is present during reinforcement?

Associative answer: The CSs compete for a share of a limited increment in associative strength; or, selective attention to one CS denies other CSs access to the associative mechanism (CS processing deficits); or, predicted USs lose the power to reinforce (US processing deficits).

Timing answer: The rate of reinforcement is partitioned among reinforced CSs in accord with the additivity and predictor-minimization constraints.

ActionTuple answer: Each CS will have a Predictor that predicts the appearance of reinforcement; thus the creature will be highly confident that reinforcement is forthcoming, and the confidence in both Predictors will rise when reinforcement appears. Predictor minimization constraints are handled by the culling sentinel, which may observe redundancy among the Predictors.

11. How does conditioned inhibition arise?

Associative answer: The omission of an otherwise expected US (the occurrence of a No-US) strengthens inhibitory associations.

Timing answer: The additive solution to the rate-estimation problem yields a negative rate of reinforcement.

ActionTuple answer: A Predictor that uses inverted Percept activations is formed. Its containing ActionTuple obtains a perceived value that indicates a negative rate of reinforcement.

12. What happens when a CS follows a reinforcer rather than preceding it?

Associative answer: Nothing. Or, an inhibitory connection between CS and US is formed.

Timing answer: A negative CS-US interval is recorded, or, equivalently, a positive US-CS interval. (More precisely: subjective intervals, like objective intervals, are signed.)

ActionTuple answer: An area for future investigation. The representations used – including signed, subjective Predictor intervals – are sufficient for representing Backward Conditioning phenomena. Either allowing the generation of negative CS-US intervals, or considering Predictors as bi-directional, would allow Timing results to be reproduced.

13. How does a secondary CS acquire potency?

Associative answer: An association forms between the secondary CS and the primary CS, so that activation may be conducted from the secondary CS to the primary CS and thence to the US via the primary association.

Timing answer: The signed interval between the secondary and primary CS is summed with the signed interval between the primary CS and the US to obtain the expected interval between the secondary CS and the US.

ActionTuple answer: Two Predictors are generated: one representing the signed interval between the secondary and primary CS, and the other representing the signed interval between the primary CS and the US. These intervals are summed to obtain the expected interval between the secondary CS and the US. The utility of the secondary CS is recursively computed using these two Predictors. (The concept of perceived value makes this possible. The confidence of the two Predictors is multiplied to obtain the expected confidence that the secondary CS will produce the US.)

14. How is CS-US contingency defined?

Associative answer: By differences in the conditional probability of reinforcement.

Timing answer: By the ratio of the rates of reinforcement.

ActionTuple answer: By ActionTuples representing the perceived confidence in a temporal relationship between the CS and the US.

15. What is the fundamental experiential variable in operant conditioning?

Associative answer: Probability of reinforcement.

Timing answer: Rate of reinforcement.

ActionTuple answer: The reliability with which one event follows another after an interval.

5.4 Cue Competition

Conditioning to one conditioned stimulus does not occur independently of the conditioning that occurs to other stimuli. Gallistel and Gibbon's model is particularly effective at explaining cue competition phenomena, which describe the interplay of different stimuli during a conditioning procedure. It would be constructive to consider how this architecture should reproduce or approximate these phenomena, even though we have not yet had the opportunity to formally reproduce these experiments.

In the following sections, we will adopt some terminology and abbreviations from the psychological literature. The *conditioned stimulus*, or *CS*, is represented in this architecture by a stimulus found in the TriggerContext or ObjectContext of an ActionTuple. The *unconditioned stimulus*, or US, is a stimulus predicted by a Predictor. Each Predictor contains one and only one US, although an ActionTuple may contain multiple Predictors and thus predict multiple unconditioned stimuli. In most experiments, the

US is "unconditioned" because its appearance is considered to be inherently rewarding or punishing; for example, the appearance of food, or a foot shock.

5.4.1 Blocking

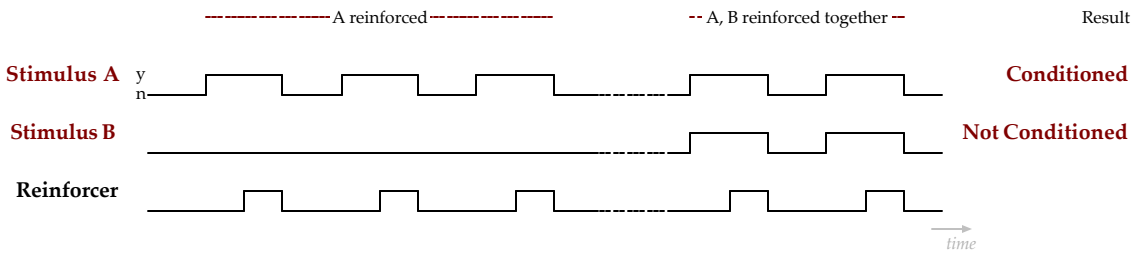


Figure 43: Blocking procedure.

Blocking describes the phenomenon that occurs when one CS is presented alone some of the time, and together with a second CS some of the time. If the rate of reinforcement during presentations of the first CS – Stimulus A in Figure 43 – is unaffected by the presence or absence of the second CS (Stimulus B), then the second CS does not get conditioned no matter how often it is reinforced.

During a blocking procedure, this architecture would first cause the creature to form the Predictor that the first CS (Stimulus A) predicts reinforcement. The creature's confidence in this Predictor would increase as repeated appearances of Stimulus A are followed by reinforcement.

After this "A-predicts-US" Predictor has been generated, the creature will have no reason to generate another Predictor the next time the US appears. It will have already been predicted by the first Predictor. Thus, in accord with the principle of parsimony, the architecture finds no reason to learn about a relationship between Stimulus B and the US, and blocking is correctly reproduced.

5.4.2 Overshadowing

When two CSs are always presented and reinforced together, a conditioned response generally develops much more strongly to one than to the other. This phenomenon is known as *overshadowing*.

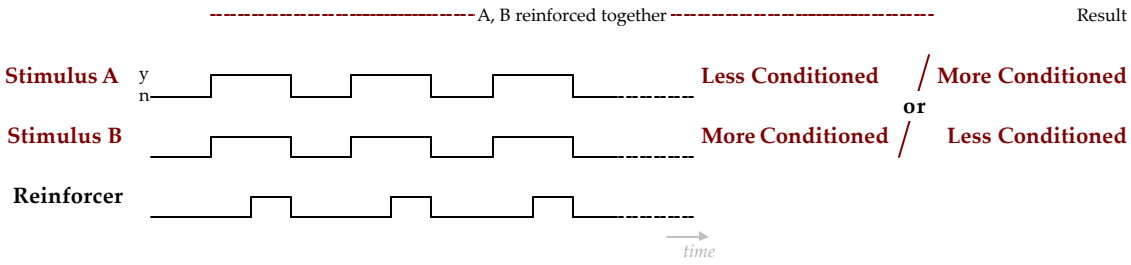


Figure 44: Overshadowing procedure.

For any combination of CSs that have always occurred together, Rate Estimation Theory allows the stimuli to register an infinite number of rate estimates that sum to the observed rate of reinforcement. The principle of predictor minimization eliminates the redundant CSs, resulting in the selection of an "overshadowing" stimulus that dominates the other stimuli and minimizes the number of stimuli credited with predictive power. The stimulus that ends up overshadowing the others seems to be often arbitrarily selected. Some work suggests that creatures may have innate biases toward selecting a particular type of stimulus [Foree, LoLordo 1973].

RET explains overshadowing and blocking without involving any free parameters for stimulus salience. While those free parameters already exist in this architecture (and have already proven valuable during action selection), our philosophy parallels RET in that the method we use for predictor minimization does *not* depend on the salience of stimuli. Instead, it relies on the culling sentinel mechanism that seeks to preserve *only the simplest explanation* for the appearance of a stimulus.

The culling sentinel, as described in Section 3.3.6, would not generate overshadowing phenomena, but it could readily be augmented to do so. It currently eliminates the more complex of a pair of equally-reliable Predictors that both predict the same outcome. The sentinel could similarly remove a Predictor if another Predictor of the same outcome already exists, and the contexts for both Predictors consist of stimuli with very high concurrence.

5.4.3 One-Trial Overshadowing

Overshadowing effects can become apparent after only one trial during which redundant CSs are reinforced. This *one-trial overshadowing* effect should also be produced by this architecture as a result of the *react* operation's attempt to isolate a single, simple explanation for the appearance of an unexpected stimulus. It generates a *single* Predictor to explain an unexplained stimulus, immediately producing overshadowing effects.

No attentional process is needed to exclude other CS candidates from "access to the associative process," as Gallistel puts it. However, when the creature needs to randomly choose between two CSs to select one explanation for use in a Predictor, it makes sense to take advantage of the creature's attentional process by weighting the decision slightly in favor of a CS found in the creature's current object of attention.

5.4.4 Relative Validity

The architecture should correctly model the *relative validity* effect, first demonstrated by Wagner and discussed in [Wagner, Logan et al. 1968]. Consider three stimuli labeled A, B, and X which are used for two types of trials: AX trials, in which A and X are presented; and BX trials, in which B and X are presented. There are two protocols: P1, for which AX trials are reinforced and BX trials are not; and P2, for which half the AX trials and half the BX trials are reinforced. Subjects that are exposed to the P1 protocol develop a conditioned response to stimulus A only, and subjects exposed to the P2 protocol develop a conditioned response to stimulus X only. Hopefully Figure 45 will help make this comprehensible.

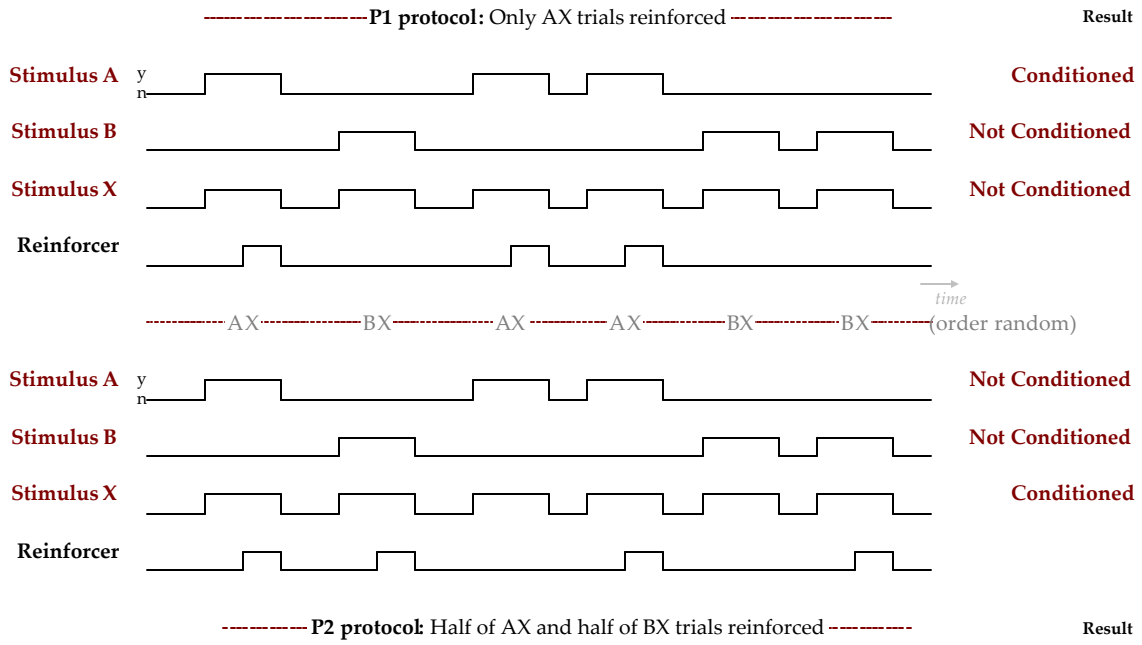


Figure 45: Relative validity procedure. In the P1 protocol, subjects have AX trials reinforced; in the P2 protocol, subjects have half the AX and half the BX trials reinforced. P1 subjects develop a response to stimulus A only; P2 subjects develop a response to stimulus X only.

The Predictor generation and culling mechanisms in this architecture should accurately arrive at the result for the P1 protocol. A creature exposed to the P1 protocol will begin by generating a Predictor with either X or A as the context. If A is chosen, no further surprises will occur. If X is chosen, the creature will experience expectation violations during each BX trial, until the Predictor is no longer a sufficiently reliable predictor to explain the reinforcer. Through Prediction refinement – or, if the first Predictor is culled, another react operation – the expected Predictor involving A will be generated.

A creature exposed to the P2 protocol will begin with one of three hypotheses: that either A, B or X results in a reward. If the creature chooses A or B, on a subsequent trial it will also hypothesize that one of the remaining two stimuli also predicts a reward. The challenging case is the one wherein the creature selects both A and B, and thus is on the road to developing a conditioned response to both A and B. In this case, the formation of the prediction that X results in a reinforcer should come from the partial reinforcement schedule, because neither A nor B will predict reinforcement with particularly high reliability. The react operation, when performed on the appearance of the reinforcer, may determine that the reinforcer was not sufficiently predicted by any existing Predictor, and generate a new hypothesis that X predicts reward.

As is the case with Overshadowing, to perfectly reproduce this phenomenon, an augmentation to the culling mechanism is needed that that would recognize A and B's concurrence relationship with X, eventually causing the first two Predictors to be culled.

5.4.5 Inhibitory Conditioning

Inhibitory conditioning describes a procedure in which the presence of a stimulus predicts the omission of reinforcement. Two features of the architecture make it possible for creatures to respond appropriately to this procedure.

First, the SET/RET model records *subjective* rates of reinforcement. In this architecture, although we do not incorporate rates of reinforcement, we record *subjective* DriveVectors which can be negative as well

as positive. Value is represented in terms of those DriveVectors, which contain a perceived effect on the creature's drives *relative to how they would change otherwise.* Thus, if the creature predicts that in the absence of action it will feel increasing amounts of pain, but in the presence of action A its level of pain will remain constant, then the DriveVector that will come to be associated with action A is not zero but rather very high. Similarly, if the creature predicts that a reinforcer is imminent except if a particular stimulus is perceived, then that stimulus will have a high magnitude value, representing a relative increase in drives.

The second feature of the architecture that makes this possible is that the creature is able to represent the absence of a Percept as easily as it can represent its presence. It is a well-established fact that animals do not represent the absence of a stimulus as readily as they represent its presence (Gallistel, pers. comm.), an effect that we could model using different levels of inherent salience for each of the stimuli. But regardless, both "absence" and "presence" Stimuli can potentially considered interesting by the TimeRate System (see "Finding Stimuli in existing Perceptual Representations," Section 4.1.1).

We note that in this formulation, as in Gallistel's, the conditioned effects of an inhibitory stimulus have nothing to do with inhibition in the neurophysiological sense.

5.5 Background Conditioning

Another class of conditioning experiments it would be useful to consider is the set of *Background Conditioning* procedures. The *truly random control* protocol describes an experiment where the background rate of reinforcement is the same as the rate of reinforcement when a transient CS is also present. The important result here is that conditioning depends on the CS-US relationship, rather than simply the pairing of the CS and US.

In this architecture, truly random pairing between the CS and the US will prevent the creature from arriving at a reliable apparent temporal causality relationship. As the two stimuli continue to randomly appear, assuming they are both salient, the creature will attempt to explain their appearance and generate appropriate Predictors during the react operation. The Predictors formed will suggest that one stimulus follows the other after a particular interval described by Scalar Estimation Theory. Any such Predictor will inevitably be proven unreliable and discarded.

As a result, during the truly random control experiment, the creature will continuously be generating low-confidence expectations as it perceives the CS and US. These expectations will not be met in a sufficiently reliable way and the Predictors making those expectations will eventually be discarded. Thus this architecture should perform like a real subject during a truly random control procedure: Conditioning *does* depend on a CS-US relationship.

5.6 Backward Conditioning

Backward conditioning produces an association where the subject learns that the US precedes the CS. From a prediction point of view this is problematic, as the CS does not enable the creature to anticipate the US. But an elegant experiment by Cole et. al. demonstrates how *backward* conditioning can sometimes provide a creature with a mechanism for predicting an event *forward* in time [Cole, Barnet et al. 1995].

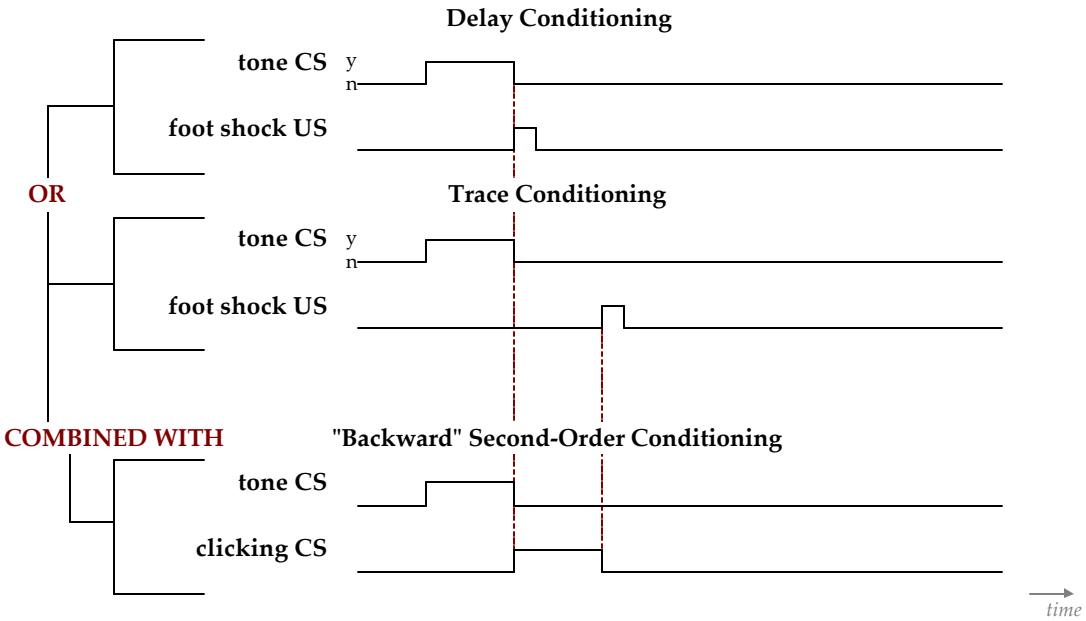


Figure 46: The Cole experiment (after [Gallistel, Gibbon 2000], Figure 22A).

There are two versions of Cole's experiment. In the first, a *delay protocol* is used, whereby a tone CS is followed after a delay by a foot shock US. In the second version, a *trace protocol* is used, in which the tone is sounded, and then stopped, and then after an interval the foot shock US appears. (see the above Figure). In both protocols, there is then a second stage of backward second-order conditioning, in which the tone CS is followed by a clicking CS. This phase must be kept brief to prevent extinction of the conditioning that occurred during the first stage.

Subjects who experience the trace conditioning protocol followed by the backward second-order conditioning show signs of fear after perceiving the clicking CS. Explanations of this phenomenon from associative and timing perspectives are found in Gallistel's paper [Gallistel, Gibbon 2000]. The architecture described here should be capable of simulating this phenomenon by generating the prediction of a tone CS backward in time, which in turn could generate a prediction of impending foot shock US forward in time (past the present), thus triggering an avoidance response to the predicted shock.

Subjects who experience the delay protocol instead show little or no fear. For those subjects, the expected interval to shock at the onset of the secondary CS (the clicker) is 0. Thus, by the time they perceive the onset of the CS, there is nothing to fear.

The TimeLine and Predictor mechanisms, as well as the implementation of the action selection mechanism, are capable of representing the prediction of an event in the past. The intuition for why this might happen is that the creature may believe it failed to *perceive* the event in the past. However, Backward Conditioning will not emerge under the current Predictor generation procedure, as all explanations it correctly created to account for the unexpected appearance of a stimulus look *backward* in time for an appropriate explanation. We propose two ways that this problem could be addressed in the Future Work section that follows.

5.7 Section summary

The cognitive architecture benefits from a tight integration of Scalar Expectancy Theory (Section 5.1). We have developed heuristics that let us incorporate the principles behind Rate Estimation Theory, and discussed challenges inherent in a direct implementation of its computational model (Section 5.2). We

summarized the architecture's relationship to the existing associative and timing models of conditioning (Section 5.3).

The architecture should, in theory, correctly reproduce several cue competition phenomena (blocking, one-trial overshadowing, inhibitory), and would likely be able to reproduce others (relative validity, overshadowing) with additional augmentation (Section 5.4). The system should correctly reproduce background conditioning phenomena (Section 5.5), but a change to Predictor generation would be required to perform backward conditioning (Section 5.6). Further discussion of avenues for future work follows in Section 7.

6.0 Related Work

This work borrows heavily from the impressive work that has come before. We summarize here some of our inspirations from the fields of virtual ethology and conditioning.

6.1 Virtual Ethology

Using a simple perception-output mapping, Braitenberg ascribed to his vehicles affective qualities ranging from love to fear, representing one of the original forays into the realm he called "synthetic psychology" in [Braitenberg 1986]. Reynold's boids algorithm represented the first behavior-controlled animation; see [Reynolds 1987]. Tu and Terzopolous's physically-based artificial fish model, described in [Tu, Terzopoulos 1994], incorporated a perceptual model and a behavior system. Perlin's Improv system is designed to create interactive actors. As opposed to beginning with intelligence, Perlin is interested fundamentally in creating "actors" with powerfully scripted behaviors [Perlin, Goldberg 1996]. Damasio's somatic markers, described in [Damasio 1995], are a precursor to our drive-based value attribution. A number of recent commercial software products have focused on interactive characters, including PF-Magic's Dogz series [PF.Magic 1996], Cyberlife's Creatures series [Cyberlife 1998] and Evans' remarkable titans in Black and White, which he describes in [Evans 2001]. Our Predictor innovation mechanism is functionally analogous to his entropy-based dynamic decision trees. The Synthetic Characters Group's system architecture, known as c4, is described in [Isla, Burke et al. 2001], and in excruciating detail in [Burke, Isla et al. 2001].

The importance of considering perception and learning together was emphasized by Barlow in [Barlow 1990], in which he concludes that perception must play an important role in providing a representation that promotes the efficient learning of predictive associations. Kline provides a discussion of prediction for synthetic characters, and discusses the differences between surprise and expectation violation in [Kline 1999]. Maes and Drescher also provide insight into working with reliability in [Maes 1989] and [Drescher 1991]. Allen's work on temporal logic integrates temporal reasoning into a planning system [Allen 1991].

deKleer provides a solid introduction to causal theories in [deKleer, Brown 1986]. Further discussion of the application of causality is found in [Iwasaki, Simon 1986] and [deKleer 1986]. Pearl's discussion of how we try to "explain away" why an event occurred in [Pearl 1988] influenced this architecture's explain operation. Sheridan discusses the purpose of cognition and mental models in [Sheridan 1992], and also describes behaviorist and hermeneutic challenges to mental models and rationality in cognitive science. Also see Moray for more on the structure of mental models and the different types of causality in [Moray 1990].

Finally, the structure of this thesis is largely inspired by Blumberg's Ph.D. work, [Blumberg 1996], in which he describes how the architecture of the Alive project synthesized ethological principles and classical animation. In this thesis, we similarly detail a model inspired by ethology, provide a robust implementation in a creature, and discuss how that virtual creature is able to emulate many phenomena observable in live subjects.

The shoulders of giants, indeed.

6.2 Models of Conditioning

As described in Section 5, Gallistel and Gibbon's timing model, detailed in [Gallistel, Gibbon 2000], contrasts sharply with the standard model of conditioning mathematically formalized by Rescorla and Wagner in [Rescorla, Wagner 1972] and [Wagner, Rescorla 1972], and described by Domjan in [Domjan 1998]. Using a nonstationary, multivariate time series analysis, Gallistel developed a spreadsheet model of conditioning that incorporated SET and RET, available as [Gallistel 1992].

7.0 Concluding Remarks

7.1 Important Ideas

7.1.1 Causality and Action Selection are integrated

While it may require a leap of faith for cognitive psychologists to assume that animals can implement a subjective timing mechanism with neurons, computer scientists require no such leap, just a few lines of code. But although it was easy code Scalar Expectancy Theory's timing mechanisms into Predictors, it was much more difficult to figure out how the action selection mechanism could make use of the expectations those mechanisms were generating.

The single most important realization that allowed us to take advantage of causality information was that the contexts a creature uses to trigger actions can be equivalent to the contexts they use to trigger predictions. If we use the same structure for both (in our case, the context of an ActionTuple), we reap tremendous benefits, in that the creature can easily integrate knowledge of how the world works into its action selection.

If a creature would like an event to occur in the future, it can perform an action that causes the expectation that that event will occur after a given interval. Even if the appearance of some *stimulus* should apparently cause a prediction of some useful future event, the creature can look for strategies that result in the appearance of that *stimulus*. From an affective point of view, a creature can attribute a value to each stimulus based on how its onset can help move the world toward a more desirable state.

7.1.2 Attention Selection and Action Selection are integrated

The previous Synthetic Characters architecture distinguished between an *attention selection* mechanism and an *action selection* mechanism. We have found it useful to tightly integrate the two in the new architecture.

The old attention selection mechanism employed a series of heuristics to set the object of attention. The action selection mechanism could override that decision if it believed it had a better idea of what the creature should attend to. The decisions made by the attention selection mechanism, however, did not influence the action selection mechanism.

The heuristics used to guide the attention selection mechanism – the observation of things that are large, moving fast, suddenly appear, and so on – are precisely the kind of events that the react operation is called upon to process in the new architecture. Each of these events should have an effect on the creature's object of attention, and in addition, each should also cause the creature to consider interrupting its current behavior in order to provide a more involved response. A change in the creature's focus of attention could be considered one part of such a response.

We also note that the target object chosen by the explore operation is influenced by the current object of attention. Thus the action selection mechanism takes advantage of attention selection results.

7.1.3 A Desire to Understand the World Drives Learning

Our fundamental assumption is that learning is driven by a creature's desire to *understand its world*. When an event occurs that the creature doesn't predict, the creature is *surprised* and invents an explanation for why the surprising event occurred. When an explanation (in the form of a Predictor) turns out to be erroneous, an *expectation violation* occurs and the creature either refines the explanation or invents a new one. In the absence of unusual stimuli, *a creature's curiosity drive* motivates it to explore the world. Thus, it bears repeating that all three fundamental motivations for learning emerge from a desire to *understand the world*.

Certainly, the creatures described here are highly motivated to satisfy their drives by obtaining reinforcement from the world. An attempt to maximize rate of reinforcement (or value) is fundamental to many existing architectures and learning techniques. But in this case, instead of motivating a creature's learning with a perpetual attempt to maximize rate of return, these creatures instead seek to understand *enough* about the world to satisfy their drives effectively and predict the onset of salient events. They are *then* motivated by curiosity to discover new things, some of which may lead to new techniques for maximizing rate of return. One observer called this the "curious slacker" approach.

The results suggest it creates creatures that are better able to sustain the illusion of life. Perhaps we ourselves are curious slackers. Or perhaps it's just me.

7.1.4 The Cognitive Economy

For every source there must be a sink. For every mechanism that deposits topology, there must be a mechanism that performs withdrawals. The architect of a brain must consider these issues when thinking about the performance of the system on various time scales – eight seconds, eight minutes, eight hours, eight days – as well as in the theoretical limit. What will happen, for example, to knowledge that is rendered useless by a change in the environment?

When virtual financial economies reach a critical mass, they predictably behave like their real-world equivalents, and their management is wrought with similar challenges, such as those described by Simpson in [Simpson 2000]. We have sought to address many of these challenges in the cognitive economy already: over- and underproduction of commodities (like ActionTuples, Predictors, Beliefs, and Percepts for classifiers) causing deflation and inflation accordingly (the value of learning an important bit of knowledge relative to the computational cost of that knowledge); an unwillingness of participants in the economy to use the available sinks (for example, ActionTuples refusing to be removed by the culling sentinel); and above all, the tremendous challenge presented to an "economist" who seeks to understand and predict the behavior of the system at macro- and micro-economic levels.

Animal brains are necessarily a sort of cognitive economy, as they are restricted by the finite number of neurons that can fit into the brain skull cavity. An intriguing recent study suggests that licensed London taxi drivers have significantly larger posterior hippocampi. Hippocampal volume was found to correlate with the amount of time spent as a taxi driver. The researchers conclude there is a capacity for local plastic change in the structure of the adult human brain in response to environmental demands. [Maguire, Gadian et al. 2000] That the human brain can accommodate such change without compromising the functionality of other systems is astonishing. Perhaps we need to develop an agent or group of agents like the "B-Brain" watchdog proposed by Minsky (see [Minsky 1985]) that would similarly accommodate changes to a virtual brain in response to environmental demands.

7.1.5 Nothing is Deterministic, and Many Distributions aren't Linear

An important part of designing a system that behaves reasonably like a real creature is coming up with fitting (and hopefully intuitive) probability distributions for many of the operations that the system needs to perform. Every Selection in Section 3.2.3 provides an example – Action Selection, Drive Selection, Strategy Selection, and so on.

It is almost always the case that when a system in a virtual creature makes a deterministic decision based on the "best option," the creature is afforded an opportunity to get stuck in a mindless loop. It invariably will. Even during operations like *exploit* that are meant to produce the "best" option, each decision should always employ some degree of randomness.

That being said, it's rare that a linear histogram probability distribution produces the desired results. There is evidence to suggest that in some selection processes, animals behave like ideal detectors, dividing their time between two or more behaviors in such a way as to maximize the reward provided by the various options (see [Gallistel, Mark et al. 2001]). But in other kinds of selections, such as the one performed by the DogEar Utterance Classifier when it classifies a new utterance, a linear probability

distribution produces frustrating results (see Appendix A). If the match metric between an utterance and the "sit" group is 0.4 out of 1.0, the match between the utterance and the "down" group is 0.5 of 1.0, and the match for all other groups negligible, we should want to select "down" much more often than 5/9 ? 56% of the time! On the other hand, under these conditions, we should want to select "down" somewhat *less* than 100% of the time, or we may never explore the possibility that the utterance is being misclassified.

Time spent choosing an optimal and intuitive distribution for a selection process is rarely wasted. The theory of probabilities, noted Laplace, "is nothing more than good sense confirmed by calculation."

7.1.6 A Good Visualizer Is Worth Thousands of Lines of Debug Spew

The Percept Tree, Autonomic Variable, Action Selection, TimeLine and other visualizers proved indispensable during the creation of this architecture. Many problems were diagnosed and solved as a result of watching the creature in action and concurrently monitoring the state of its brain. For example, the need for an affective model that responds appropriately to predictions, expectation violations, and explanations (see Figure 25 and Section 3.4) became apparent while observing the results of an earlier, more naïve affective model.

The author marvels at Tufte's capacity to visualize information, although he can only aspire to his elegance (both [Tufte 1990] and [Tufte 2001] are highly recommended). We offer two important lessons here. First, the visualization apparatus should be implemented as an *entirely* separate entity from the rest of the architecture. Second, low-pass filters and other embellishments that alter the information being presented are entirely unwelcome in the display. Such techniques reduce the amount of information presented, and cause uncertainty about whether an effect comes from the underlying process or the visualizer, whose function should be to effectively convey information.

7.1.7 The World Resists Oversimplification (Beyond Simple Credit Assignment)

Many credit assignment and machine learning algorithms make substantial assumptions about how the world is represented. Q-Learning, for example, requires that the world is divided into discrete states, transitions between those states and is always only in one state.

There is a large class of problems for which this technique is demonstrably useful (see [Kaebling, Littman et al. 1996] for a survey). But even the relatively simple virtual worlds described here resist reduction into simple states. The real world isn't ever in a single simple state, nor does it conveniently transition from a single state to another single state. If we are overzealous in our attempts to simplify our mental representations of the world, we risk introducing what McCallum calls *aliasing* – the inability for learning representations in the system to learn the right things, because the perceptual representations can't distinguish between the things they need to learn about (see [McCallum 1995]).

Credit assignment in this architecture is guided by apparent temporal causality, but many of the causality relationships doesn't fall nicely into the category "A obviously precedes B, thus A predicts B." Some, but not all, are a result of self-action. Learning is guided by temporal proximity, but also salience, existing knowledge, and common sense about how the world works. In summary: the world is not a simple place, and our representations and learning algorithms must embrace this.

7.2 Summary of Contributions

We summarize here the key implementation and functionality details that distinguish this framework from the previous Synthetic Characters architecture.

7.2.1 Implementation

- *All value in the system is subjective and drive-based.* Subjective values are fundamental to the time and rate representation found in [Gallistel, Gibbon 2000], and multidimensional drive-based

values have been re-incorporated from their use in the Alive project, described in [Blumberg 1996].

- *The TimeLine* provides a convenient collection of salient events both perceived and predicted in the past, present and future.
- *The Stimulus representation and TimeRate System* provide useful abstractions for maintaining statistics and filtering salient perceptual information that will be processed by the action selection mechanism.
- *Predictors* provide a means for interacting with the TimeLine to represent knowledge of apparent temporal causality.
- *A Results augmentation to Blumberg's existing ActionTuple representation* that integrates causality into the action selection representation.
- *A new attention- and action-selection mechanism* with explore, exploit, react and startle as the fundamental operations allows the creature to perform coherent, relevant actions and generate appropriate affective responses to perceptions and predictions.

7.2.2 Functionality

- *The ability to predict future events and react to those events,* thus integrating elements of reactive and planning systems.
- *The ability to discover and refine knowledge of causality relationships* in the world which may or may not involve self-action by a process of hypothesize, test, refine.
- *The reproduction of a variety of new conditioning phenomena,* including blocking, overshadowing, and other cue competition phenomena, as well as generalization and discrimination.
- *An affective model that produces emotional memories* about how things in the world – objects, creatures, actions, and so on – affect the creature's drive state, facilitating a utility metric that is a function of the creature's current drive state.
- *The ability to perform reinforcement learning,* even when a reinforcer does not immediately follow the action being rewarded, by a process of learning new reward states and generating perceived values.

7.3 Future Work

7.3.1 Further integration of rate information

A fundamental difference between this system and the representations suggested by Rate Estimation Theory is that this architecture places much more weight on *events* such as the onset and offset of stimuli. The creature does not make predictions based on the sustained presence of a stimulus. Thus, the architecture is adept at modeling causality relationships between discrete "events." It is also effective at modeling rates of reliability. But it is not yet capable of modeling rates of reinforcement in the sense described by Rate Estimation Theory.

One problem with this architecture's emphasis on Predictors is that their assumption of "trials" is very awkward for events that are generated by a random rate process, and thus do not regularly occur a fixed interval after the onset of some context (the Predictor Context). When events are generated by a random rate process, the creature ends up generating multiple, low-confidence Predictors that predict the appearance of the stimulus after varying interval lengths. It would be better if the representation actually encoded in a single Predictor the concept "an event will occur with a given (low) probability at some point in the next while, but I can't be much more precise about the interval." One way to "retro-fit" this functionality into the existing architecture would be to have a sentinel detect the existence of

many similar Predictors with differing interval lengths, and amalgamate them into such a "random rate" Predictor.

7.3.2 Integration of other Explanations

Concurrent work by Isla on spatial competence will provide the action selection mechanism with new explanations for expectation violations that arise from spatial common sense (see [Isla 2001]). For example, occlusion may explain the creature's inability to perceive a predicted stimulus. Integrating Isla's work into the architecture would provide an intriguing testbed for merging knowledge of temporal causality and common sense.

7.3.3 Grouping of ActionTuples

One way the system won't scale is that innovation eventually results in an unmanageable number of ActionTuples that reside in one big list in the action selection mechanism. We need some way to partition them so that they are perhaps task-oriented, or at least easier to manage and search through. Minsky notes (in [Minsky 1985]) that the human mind's ability to recall useful information when it is needed without being swamped with useless information requires an effective way to organize our memories. Unfortunately for those of us trying to implement such a mechanism, the techniques our own minds use to perform this task are inaccessible to consciousness.

7.3.4 Long-Term Memory

The creature's Working Memory currently contains *Beliefs* which represent caches of Percept activation data for the various objects in the world. Each of the Percepts acts like a feature, and the Belief provides the object representation. When the creature does not perceive or predict the presence of an object for an amount of time, its corresponding Belief is eventually culled from Working Memory. But we would like to learn about the predictive properties of these *particular* objects in the world!

At a more fundamental level, this architecture lacks a long-term semantic memory that would allow it to store and recall objects the creature has perceived in the world. Such a system would allow it to manage the Beliefs in Working Memory, possibly associating them with long-term concepts like "my friend the shepherd" and "the shed beside the shepherd's dwelling" in a semantic memory. The persistence of a concept about these objects beyond their stint in Working Memory could allow the creature to learn about objects on a more permanent basis, perhaps by providing the TimeRate System with Stimuli based on concepts on Long-Term Memory.

7.3.5 Learning Higher-Level Goals and Concepts

An exciting future direction would be to extend the learning mechanisms so that a creature can discover – and then learn how to satisfy – higher-level goals. For example, in the Trial By Eire installation, Duncan the terrier can learn that the shepherd wants him to circle the sheep clockwise, but he can't understand that the shepherd's intention is to move the sheep south down the field. To understand an intention like this, Duncan would need to represent more abstract, high-level changes to the world. In the Goatzilla domain, it would be exciting for that creature to learn about and practice resource management. In his current state, at the rate he's going through sheep, it's unlikely he'll survive the harsh Highland winter. More abstract goals and concepts would facilitate many exciting new kinds of behavior.

7.3.6 Theory of Mind

The creatures in the current system have no theory of mind. Although many ethologists would argue it's unlikely that dogs are capable of theory of mind (see [Shettleworth 1998] for discussion), integrating theory of mind would be an exciting avenue for future research.

Goatzilla kicks the shed because he knows that action will eject the sheep, but he has no concept of the fact that they're running because they're scared out of their wits. Without theory of mind, he can learn

that kicking the shed causes a loud noise and makes the shed rumble, and the sheep bolt as a result. But what if he could learn that kicking the shed causes a loud noise and a rumble, and that terrifies the sheep, who end up bolting *out of fear?* Understanding even a little about the sentience of other creatures would give the learning mechanisms new insights into which strategies and avenues for exploration might prove effective in the future. For example, if a creature wants to make sheep bolt, and he knows they bolt when they're afraid, that creature might ask himself: well, what would make *me* afraid?

Theory of mind would assist in understanding *final causality* as described by Moray in [Moray 1990], and discussed previously in Section 3.3. The notion of integrating the other types of causality – material causality, efficient causality and final causality – to this system's understanding of formal causality is both exciting and daunting.

7.3.7 Augmented Predictor Generation: Playing with Cause and Effect

Backward Conditioning is not possible under the current scheme, as all explanations generated by the *explain* function to explain the onset of a stimulus look backward in time for a cause *(cause*, it assumes, *precedes effect).* We propose two untested techniques for augmenting Predictor generation that might allow for effects like Backward Conditioning.

The first would be to consider every Predictor to be bi-directional; that is, we employ the knowledge that the effects of a Predictor may have been caused by the earlier appearance of that Predictor's context. Thus, when the react operation is called upon to explain the appearance of a stimulus, if it fails to find a Predictor on the TimeLine that explains the event, it could look for Predictors that *would have* reliably predicted the event had they been previously activated. It is possible that the creature simply did not perceive the activation of a Predictor Context; and if it had, then it would have predicted the appearance of this stimulus. A significant difficulty with this approach would be determining which, if any, of the possible causes we should attribute to an effect.

The second way to perform Backwards Conditioning would be to generate Predictors that specifically predict the prior appearance of a stimulus. (Since the interval encoded is already relative, we are able to represent this as a "negative" interval.) It remains unclear why the creature would want to form such a Predictor. Why generate the Predictor that A precedes B instead of the more useful one that says B follows A? Perhaps because A predicts the reinforcer C.

7.3.8 Negative Knowledge and the Culling Sentinel

When is negative knowledge useful and when is it simply a waste of memory? The culling sentinel could also be improved to implement some of the protocols discussed above if it employed knowledge from the RET temporal correlation matrix that is now stored in the TimeRate System. The most obvious implementation of the Overshadowing protocol, for example, requires the culling of ActionTuples that are identified as redundant because their start contexts exhibit a high degree of temporal correlation. Such a mechanism would also make the architecture's ability to model the P2 protocol of Relative Validity much more reliably (see Section 5.4.4).

7.3.9 Spontaneous Recovery and the Culling Sentinel

Two major reasons why the culling sentinel exists to remove Predictors and ActionTuples are that they take up space, and that a superfluity of ActionTuples slows down the action selection mechanism. We have already described above (see Section 7.1.4) how Predictors that were once useful might be "retired" without being culled completely. Storing the maximum reliability ever achieved by a retired Predictor might help us implement a spontaneous recovery mechanism if that Predictor was perceived to become useful again in the future.

8.0 List of References

Aittokallio, T., M. Gyllenberg, et al. (2000). Testing for Periodicity of Signals: An Application to Detect Partial Upper Airway Obstruction during Sleep, Turku Centre for Computer Science.

Allen, J. F. (1991). Planning as Temporal Reasoning. The Second International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, MA, Morgan Kaufmann.

Barlow, H. (1990). "Conditions for Versatile Learning, Helmholtz's Unconscious Inference, and the Task of Perception." Vision Research **30**(11): 1561-71.

Blumberg, B. M. (1996). Old Tricks, New Dogs: Ethology and Interactive Creatures. Media Lab. Cambridge, MIT.

Braitenberg, V. (1986). Vehicles : Experiments in Synthetic Psychology.

Brooks, R. A. (1991a). Intelligence without Reason, Computers and Thought lecture . IJCAI-91, Sidney, Australia.

Brooks, R. A. (1991b). "Intelligence Without Representation." Artificial Intelligence Journal **47**: 139-159.

Burke, R. C., D. A. Isla, et al. (2001). Creature Smarts : The Art and Architecture of a Virtual Brain . Game Developers Conference 2001, San Jose, CA.

Cole, R. P., R. C. Barnet, et al. (1995). "Temporal encoding in trace conditioning." Animal Learning and Behavior **23**(2): 144-153.

Cyberlife (1998). Creatures 2.

Damasio, A. (1995). Descarte's Error, Harvard University Press.

deKleer, J. (1986). "An Assumption-based TMS." Artificial Intelligence Journal **28**(2): 127-162.

deKleer, J. and J. Brown (1986). "Theories of Causal Ordering." Artificial Intelligence Journal **29**(1): 33-62.

Domjan, M. (1998). The Principles of Learning and Behavior.

Downie, M. (2001). Behavior, Animation and Music: The Music and Movement of Synthetic Characters. The Media Lab. Boston, MIT.

Drescher, G. L. (1991). Made -up minds: a constructivist approach to artificial intelligence. Cambridge, Mass., MIT Press.

Ekman, P. (1982). Emotion in the Human Face . Cambridge, UK, Cambridge University Press.

Evans, R. (2001). The Future of AI in Games: A Personal View. Game Developer. **8:** 46-49.

Foree, D. D. and V. M. LoLordo (1973). "Attention in the pigeon: Differential effects of food-getting versus shock-avoidance procedures." Journal of Comparative and Physiological Psychology **85**: 551-558.

Fox, M. W. (1971). Behavior of Wolves, Dogs and Related Canids . Malabar, Florida, Krieger Publishing Company.

Gallistel, C. R. (1990). The Organization of Learning. Cambridge, MA, Bradford Books / MIT Press.

Gallistel, C. R. (1992). "Classical conditioning as a non-stationary, multivariate time series analysis: A spreadsheet model." Behavior Research Methods, Instruments, and Computers **24**(2): 340-351.

Gallistel, C. R. and J. Gibbon (2000). "Time, Rate and Conditioning." Psychological Review **107**: 289-344.

Gallistel, C. R., T. A. Mark, et al. (2001). "The Rat Approximates an Ideal Detector of Changes in Rates of Reward: Implications for the Law of Effect.".

Intel, I. (1998). Intel Recognition Primitives Library v4.0 Documentation. Santa Clara, CA, Intel Corporation.

Isla, D. A. (2001). The Virtual Hippocampus: Spatial Common Sense for Synthetic Characters. MIT Department of Electrical Engineering and Computer Science. Cambridge, MIT.

Isla, D. A., R. C. Burke, et al. (2001). A Layered Brain Architecture for Synthetic Characters. IJCAI, Seattle.

Ivanov, Y., B. M. Blumberg, et al. (2000). EM For Perceptual Coding and Reinforcement learning Tasks. 8th International Symposium on Intelligent Robotic Systems, Reading, UK.

Iwasaki, Y. and H. Simon (1986). "Causality in Device Behavior." Artificial Intelligence Journal **29**(1): 3-32.

Johnson, M. P. (1999). Multi-Dimensional Quaternion Interpolation. Cambridge, MIT Media Lab.

Johnson, M. P. (2001). Quixote: Quaternion-Based Techniques for Expressive Interactive Character Animation. Media Lab. Cambridge, MIT.

Kaebling, L. P., L. M. Littman, et al. (1996). "Reinforcement learning: a survey." Journal of Artificial Intelligence Research **4**: 237-285.

Kline, C. (1999). Observation-based Expectation Generation and Response for Behavior-based Artificial Creatures. Media Lab. Cambridge, MIT.

Lindsay, S. R. (2001a). Adaptation and Learning, Iowa State University Press.

Lindsay, S. R. (2001b). Etiology and Assessment of Behavior Problems, Iowa State University Press.

Lorenz, K. and P. Leyhausen (1973). Motivation of Human and Animal Behavior: An Ethological View. New York, Van Nostrand Reinhold.

Ludlow, A. (1976). "The Behavior of a Model Animal." Behavior **58**.

Maes, P. (1989). The Dynamics of Action Selection. International Joint Conference on Artificial Intelligence, Detriot, Morgan Kaufmann.

Maguire, E. A., D. G. Gadian, et al. (2000). "Navigation-related structural change in the hippocampi of taxi drivers." Proceedings of the National Academy of Sciences **97**(8): 4398-4403.

McCallum, A. K. (1995). Reinforcement Learning with Selective Perception and Hidden State. Department of Computer Science. Rochester, New York, University of Rochester**:** 157.

McFarland, D. (1993). Animal Behavior. Harlow, UK, Longman Scientific and Technical.

Minsky, M. (1985). The Society of Mind. New York, Simon and Schuster.

Moray, N. (1990). "A lattice theory approach to the structure of mental models." Phil. Trans. R. Soc. Lond. **B**(327): 577-583.

Norman, D. A. (1993). The Power of Representation. Things That Make Us Smart. Reading, MA, Addison-Wesley.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, California, Morgan Kaufmann Publishers.

Perlin, K. and A. Goldberg (1996). "Improv: A System for Scripting Interactive Actors in Virtual Worlds." Computer Graphics **29**(3).

PF.Magic (1996). Dogz: Your Computer Pet.

Pryor, K. (1999). Don't Shoot the Dog: The New Art of Teaching and Training, Bantam.

Rabiner, L. and B. Juang (1993). Fundamentals of Speech Recognition. New York, N.Y., Prentice Hall.

Rescorla, R. A. and A. R. Wagner (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. Classical Conditioning II: Current research and theory. A. H. Black and W. F. Prokasy. New York, Appleton-Century-Crofts**:** 64-99.

Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavior Model. Siggraph 87.

Rose, C. F., M. Cohen, et al. (1999). "Verbs and Adverbs: Multidimensional Motion Interpolation." IEEE Computer Graphics and Applications **18**(5).

Russell, J. (1980). "A circumplex model of affect." Journal of Personality and Social Psychology **39**: 1161-1178.

Sheridan, T. (1992). Telerobotics, Automation, and Human Supervisory Control. Cambridge, MIT Press.

Shettleworth, S. J. (1998). Cognition, Evolution and Behavior. New York, Oxford University Press.

Simpson, Z. B. (2000). The In-Game Economics of Ultima Online. Computer Game Developer's Conference, San Jose, CA.

Spier, E. (1997). From Reactive Behaviour to Adaptive Behaviour: Motivational Models for Behavior in Animals and Robots. Oxford, Oxford University**:** 99.

Thomas, F. and O. Johnson (1981). The Illusion of Life: Disney Animation. New York, Oxford University Press.

Thorndike, E. (1911). Animal Intelligence. Darien, CT, Hafner.

Tomlinson, B. (1999). Interactivity and Emotion through Cinematography. Media Lab. Cambridge, MIT.

Treisman, A. (1998). The Binding Problem. Findings and Current Opinion in Cognitive Neuroscience. L. Squire and S. Kosslyn. Cambridge, MIT Press.

Tu, X. and D. Terzopoulos (1994). Artificial Fishes: Physics, Location, Perception, Behavior. Siggraph.

Tufte, E. R. (1990). Envisioning Information. Cheshire, Connecticut, Graphics Press.

Tufte, E. R. (2001). The Visual Display of Quantitative Information. Cheshire, Connecticut, Graphics Press.

Wagner, A. R., F. A. Logan, et al. (1968). "Stimulus selection in animal discrimination learning." Journal of Experimental Psychology **76**(2): 171-180.

Wagner, A. R. and R. A. Rescorla (1972). Inhibition in Pavlovian conditioning: Application of a theory. Inhibition and Learning. R. A. Boakes and M. S. Halliday. London, Academic Press.

Wilkes, G. (1994). Behavior Sampler, C&T Publishing.

Wilkes, G. (1995). Click and Treat Training Kit Version 1.2. Mesa, AZ.

Yoon, S.-Y., B. M. Blumberg, et al. (2000). Motivation Driven Learning for Interactive Synthetic Characters. Autonomous Agents 2000.

Yoon, S.-Y., R. C. Burke, et al. (2000). "Interactive Training for Synthetic Characters." AAAI **2000**.

Appendix A: Classification Techniques

Utterance Classification: DogEar

The DogEar system integrated into the UtteranceClassifier Percept mediates verbal communication between a human participant and one of the virtual creatures. It converts a human participant's utterance data into a Cepstral coefficient format that the creature processes. The integration of the classifier into the Perception and Action Systems demonstrates one way in which the results of action selection can influence perceptual categories.

Sound bites (recorded at 11025Hz) are obtained by using a thresholding algorithm that averages the signal over windows of 512 samples. Recording starts when the signal is above the threshold, and ends when it has been below the threshold for three successive windows. The sample is then trimmed at each end to the nearest zero-crossing.

We have chosen to use a vector of Cepstral coefficients as a representation. Inspired by the way a dog interprets the sound it hears, the creature does not comprehend language, nor does it have a concept of language. What matters is the acoustic pattern of the speech signal; thus, a Cepstral coefficient representation is sufficient to encode the necessary information. Cepstral analysis is a technique that removes the pitch ripple from high-resolution speech spectra, as examined by Rabiner and Juang in [Rabiner, Juang 1993]. The goal of Cepstral analysis is to obtain the vocal tract response after removing the pitch ripple.

The DogEar system performs this task robustly, even in high-noise environments, by filtering the log-magnitude of the signal with an inverse FFT. This is followed by truncation of the coefficients beyond the pitch frequency, and then a forward FFT [Intel 1998]. In the Fourier domain, we use 10 filters placed linearly on a scale from 100 Hz up to 2 kHz, followed by 10 additional filters laid out on a Mel scale up to 6400 Hz. Analysis is performed using a window size of 512 samples and an overlap of 256 samples per window. Dynamic Time Warping, as described in [Rabiner, Juang 1993] and [Intel 1998], is used to implement the distance metric between two utterances.

We have used two previous methods to group the utterances obtained in this way. In the incarnation described in [Yoon, Burke et al. 2000], the system included a short-term memory module with a fixed number of memory cells, each of which represented a group of classified utterances. When a new utterance was heard, it was compared to the utterances in all the groups to see if the distance of the newly arrived data in any of the groups is closer than a threshold. In the Clicker version described in [Isla, Burke et al. 2001], the second incarnation of Duncan used the token associated with each action to generate groups of classified utterances.

In the new architecture described by this thesis, instead of associating utterance groups with Actions, we associate utterance groups with ActionTuples. The result is increased flexibility in the sort of token a creature can learn. For example, the creature could learn to touch a blue object upon hearing the word "blue," and touch a yellow object upon hearing the word "yellow," a representation that would have been difficult to produce in the previous system without providing special-case ActionTuples. Additionally, the new learning mechanism's ability to generate a new reward marker using reliable predictions of a reinforcer's impending appearance allows the creature more flexibility for when it sends a reward signal to the classifier.

Deciding to add a new group to the classifier is a decision of great consequence, as the performance of the classifier degrades as the number of categories it must distinguish between increases. Thus a new group is added only when the creature creates a new ActionTuple based on the UtteranceClassifier Percept. When the action selection mechanism, through an act of innovation, creates an ActionTuple that is associated with the UtteranceClassifier Percept, the UtteranceClassifier Percept spawns a new child Percept that represents a new Utterance group. An ActionTuple can then be formed based on the

new Utterance group. If all ActionTuples based on the SpecificUtterance Percept are removed from the action selection mechanism, that utterance group is destroyed.

Gesture Classification

A similar ClassifierPercept scheme is used to implement gesture classification in the system. For the details of the system, please see [Ivanov, Blumberg et al. 2000]. The implementation similarly uses a GestureClassifier Percept and its children, GestureModelPercepts, to allow the creature to interpret and learn from the input coming from a video stream.

Appendix B: Mathematics

We summarize here many mathematical details of the mechanisms described in Section 3.

Scalar Expectancy Theory

First, we restate Scalar Expectancy Theory's timing equation (see Section 2.4.1.2):

$$t^* = k^* t_T \tag{7}$$

where

t^* is the *recorded interval*

k^* is the *timing error*

t_T is the *cumulative subjective time recorded by the timing mechanism*

Drives and DriveVectors

Evaluation of a drive d_n (see Section 3.1.1):

$$d_n(t) = \left| eval_n(t) - setpoint_n \right| \tag{8}$$

where

$d_n(t)$ is the *evaluation of drive n at time t*

$eval_n(t)$ is the *evaluation of Autonomic Variable n at time t*

$setpoint_n$ is the *set point of Autonomic Variable n.*

How Drives $d_1..d_n$ are combined into DriveVector DV:

$$DV(t) = \begin{bmatrix} d_1(t)dm_1(t) \\ d_2(t)dm_2(t) \\ \cdots \\ d_n(t)dm_n(t) \end{bmatrix} \tag{9}$$

where

$dm_n(t)$ is the *drive multiplier for drive n at time t*

Utility and Affective Stance

The utility u of something with value v (Section 3.1.1):

$$u(t) = v(t) \cdot DV(t) \tag{10}$$

where

$v(t)$ is the *value of the thing at time t*

$DV(t)$ is the *evaluation of Autonomic Variable n at time t*

$\cdot$ is the *dot product operator*

Note that for ActionTuples, the perceived value $pv(t)$ is substituted for $v(t)$ in equation (10).

The affective stance a toward something with utility u (Section 3.4) is:

$$\mathrm{a}(t) = k\,\mathrm{u}(t) \tag{11}$$

where

$u(t)$ is the *utility of the thing at time t*

k is *some constant*

Predictors: Reliability

The Long-term Reliability of a Predictor, R, is computed (as in equation (3), Section 3.3.3.3) as

$$R_{predictor} = \frac{g_T + e_T}{g_T + e_T + b_T} \tag{12}$$

where

g_T is the *number of successful Trials*

e_T is the *number of explained Trials*

b_T is the *number of failed Trials*

The Short-term reliability is computed similarly, but only takes into account the seven most recent trials.

Predictor innovation occurs when the difference between the long-term and short-term reliability metrics exceeds the innovation threshold, 0.25.

Within a Predictor, the reliability of a stimulus a, S_a, that may become part of the starting context (reproduced from equation (5)) is computed as

$$S_a = \frac{1}{2(g_T + b_T)}\left[2g_a + 1[g_T + b_T - (g_a + b_a)] + 0(b_a)\right] \tag{13}$$

where

g_T is the *number of successful Trials*

b_T is the *number of failed Trials*

g_a is the *number of times stimulus a was present during a successful Trial*

b_a is the *number of times stimulus a was present during a failed Trial*

When selecting a new Predictor context, the predictors compete probabilistically on the basis of their stimulus reliability values S (equation (13)). They are each processed by the following Boltzmann distribution function before being added to a histogram for selection of a single new stimulus to be added to the Predictor Context.

$$f(i) = e^{kS_i} \tag{14}$$

where

k is *the Boltzmann constant*

S_i is the *reliability metric value for stimulus* i (equation (13)).

Predictors: Trials and Interval Learning

The interval update equation (Section 3.3.3.2, equation (2)) that updates the recorded interval in a Predictor upon a successful Trial is

$$i_n = i_{n-1}(kR_{predictor}) + t^*(1 - kR_{predictor}) \tag{15}$$

where

i_n is the *new interval length*

i_{n-1} is the *previous interval length*

t^* is the *perceived interval of this Trial (as in equation* (7))

$R_{predictor}$ is the *Reliability of this predictor (as in equation* (12))

k is a *constant slightly less than 1.*

A Trial (Section 3.3.2) predicts an event to occur between the times

$$b_1 i_{predictor} \le t_e \le b_2 i_{predictor} \tag{16}$$

where

$i_{predictor}$ is the *Predictor Interval in the Predictor that began this Trial*

$ß_1$ is *a creature-global constant slightly less than 1*

$ß_2$ is *a creature-global constant slightly greater than 1*

t_e is the *elapsed time since the Trial began (the Predictor Context was met)*

ActionTuples

The perceived value of an ActionTuple (reproduced from equation (6)) is calculated as

$$\mathrm{pv}_i(t) = \mathrm{v}_i(t) + k \sum_{m}^{predictors} \left[R_m \sum_{n}^{facilitatedTuples} \mathrm{pv}_n(t) \right] \quad (17)$$

where

$\mathrm{v}_i(t)$ is the *intrinsic value of ActionTuple i*

R_m is the *reliability of each associated Predictor*

$\mathrm{pv}_n(t)$ is the *perceived value of each facilitated ActionTuple*

k is a *discount factor*

In this implementation, the equation uses a maximum recursive depth of 4.

Action Selection

Exploit: The exploit operation selects a single ActionTuple to activate by taking the utility value of each ActionTuple and selecting one option out of a normalized histogram probability distribution after they are processed by the function

$$\mathrm{exploit}(i,t) = e^{(k(t)d_{curiosity}(t))pv_i(t)} \quad (18)$$

where

$d_{curiosity}(t)$ is the *magnitude of the curiosity Drive (equation* (8)*)*

$pv_i(t)$ is the *perceived value of ActionTuple i*

for each ActionTuple i. The Boltzmann constant k(t) is arbitrarily set, although it provides a useful degree of freedom for tweaking the histogram's propensity to select the "best" option. This winner-take-all selection chooses one ActionTuple to become active.

Explore: The explore operation performs several selections, each using a similar distribution.

Attention Selection is performed by taking each of the Beliefs in Working Memory and selecting one option out of a histogram probability distribution after they are processed by the function

$$\mathrm{attentionSelection}(i,t) = e^{kB_i(t)} \quad (19)$$

where

$B_i(t)$ is the *interest level in Belief i at time t.*

for each Belief i.

Drive Selection is performed by taking each of the creature's Drives and selecting one option out of a histogram probability distribution after they are processed by the function

$$\mathrm{driveSelection}(i,t) = e^{kd_i(t)} - 1 \quad (20)$$

for each Drive i. The subtracted term prevents the selection of a drive with a magnitude of exactly 0.

Action Selection is then performed using equation (18).

React: The react operation must make a decision whether or not to respond to an unexpected stimulus. When it observes an unexpected stimulus, it calculates the utility (equation (10)) of all new ActionTuples representing actions and reactions (approach, avoid, observe) facilitated by the stimulus. It selects one option out of all of these options, as well as the currently active ActionTuple (which has its utility multiplied by a factor slightly greater than 1 to encourage persistence) using equation (18).