Old Tricks, New Dogs: Ethology and Interactive Creatures

Bruce Mitchell Blumberg

Bachelor of Arts, Amherst College, 1977
Master of Science, Sloan School of Management, MIT, 1981

SUBMITTED TO THE PROGRAM IN MEDIA ARTS AND SCIENCES, SCHOOL OF ARCHITECTURE AND PLANNING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY AT THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1997

Author:...

Program in Media Arts and Sciences
September 27,1996

Certified By:...

Pattie Maes
Associate Professor
Sony Corporation Career Development Professor of Media Arts and Sciences
Program in Media Arts and Sciences
Thesis Supervisor

Accepted By:..

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

Old Tricks, New Dogs: Ethology and Interactive Creatures

Bruce M. Blumberg

SUBMITTED TO THE PROGRAM IN MEDIA ARTS AND SCIENCES, SCHOOL OF ARCHITECTURE AND PLANNING ON SEPTEMBER 27, 1996 IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY AT THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

Abstract

This thesis seeks to address the problem of building things with behavior and character. By things we mean autonomous animated creatures or intelligent physical devices. By behavior we mean that they display the rich level of behavior found in animals. By character we mean that the viewer should "know" what they are "feeling" and what they are likely to do next.

We identify five key problems associated with building these kinds of creatures: Relevance (i.e. "do the right things"), Persistence (i.e. "show the right amount of persistence), Adaptation (i.e. "learn new strategies to satisfy goals"), Intentionality and Motivational State (i.e. "convey intentionality and motivational state in ways we intuitively understand"), and External Control (i.e. "allow external entity to provide real time control at multiple levels of abstraction"). We argue that these problems can be addressed in a computational framework for autonomous animated creatures by combining key ideas from Ethology and Classical Animation. We have developed a toolkit based on these ideas and used it to build "Silas", a virtual animated dog, as well as other creatures. We believe this work makes a significant contribution toward understanding how to build characters for interactive environments, be they virtual companions in immersive story-telling environments, or interesting and adaptive opponents in interactive games, or as the basis for "smart avatars" in web-based worlds. While our domain is animated autonomous agents, the lessons from this work are applicable to the more general problem of autonomous multi-goal, adaptive intelligent systems. Based on results generated from our implementation of Silas and other autonomous creatures we show how this architecture addresses our five key problems.

Thesis Supervisor: Pattie Maes, Associate Professor of Media Technology, Sony Corporation Career Development Professor of Media Arts and Sciences

Doctoral Dissertation Committee

September 1996

Thesis Advisor:...

Pattie Maes
Associate Professor, Sony Corporation Career Development Professor of Media
Arts and Sciences, Massachusetts Institute of Technlogy

Thesis Reader:...

Alex P. Pentland
Associate Professor of Computers, Communication and Design Technology,
Massachusetts Institute of Technology

Thesis Reader:...

Peter M. Todd, Ph.D.
Research Scientist, Max Planck Inst. for Psychological Research, Center for
Adaptive Behavior and Cognition

Table of Contents

List of Figures

Acknowledgments

This thesis is dedicated to my father, Phillip I. Blumberg and my late mother, Janet M. Blumberg. My father has been a major inspiration in my life through word and deed, and it was from my mother that I inherited my love of animals.

To my wife Janie, and my children Phil and Gwen I owe a tremendous debt for putting up with a husband and dad who went back to school at age 37 “to play with hamsters at MIT”, as Gwen put it. For the record, Phil is the family expert on dogs. Many thanks to each of you for your love and support.

Pattie Maes, my advisor and friend, is one of the smartest people I know, and she has been a constant source of ideas and suggestions for which I am grateful. She was the one who originally suggested that I should focus on ethology, and encouraged me to develop Silas. She also made sure I had whatever equipment I needed. Many thanks.

Sandy Pentland has also played a pivotal role in my work for which I thank him. He too has been full of ideas and encouragement, and has been very generous in terms of his time and resources. I am constantly in awe of Sandy’s breadth of knowledge and keen insight.

Peter Todd has also had a great impact on me. Through his knowledge and probing questions he has kept me honest and has helped me wrestle more than one tricky issue to the ground. He, together with Pattie, helped develop the approach to learning described in this thesis. I must also thank his wife Anita for not giving birth on the day of my defense.

My friend and collaborator Tinsley Galyean is also owed a great deal of thanks. Tinsley and I collaborated on a good deal of the underlying architecture of Hamsterdam, and wrote a Siggraph paper together. We even had fun doing it!

I must also acknowledge my debt to Jerry Schneider of the Brain and Cognitive Science department. It was Jerry’s course on animal behavior which opened my eyes to the field of ethology. Many thanks.

I’d also like to thank my office mate Michael P. Johnson. Mike, together with Brad Rhodes have been fellow travelers on the road toward artificial life. I have learned much from each of them, and value each as friends.

ALIVE has been a major focus of my life over the past four years, and I must thank all of the folks who collaborated with me on ALIVE. They include: Trevor Darrell, Chris Wren, Ali Azarbayejani, Ken Russell, Sumit Basu, Flavia Sparacino, Mike Johnson, Brad Rhodes, Thad Starner, Lenny Foner, Alan Wexelblat, Johnny Yoon, Cecile Pham, Liz Manicatide and Mike Hlavac. To each of you, many thanks for late nights and great work.

And lastly, I’d like to remember Stowie, my demented little dog. Stowie died this summer after a long and happy life and we all miss her.

Old Tricks, New Dogs: Ethology and Interactive Creatures

Bruce M. Blumberg, MIT Media Lab

1.0 Introduction

Despite many advances in the last 10 years, most autonomous agents are best characterized as “idiot savants” which typically address a single goal such as low level navigation or information retrieval, albeit robustly, in dynamic and unpredictable environments. Moreover these agents typically have only few degrees of freedom (e.g. in the case of a robot, “turn-left”, “turn-right”, “halt”) which need to be controlled. If these agents learn from their experience, once again, the learning is limited to the problem of achieving a single goal.

The use of autonomous agents in computer graphics represents a case in point. Since Reynold's seminal paper in 1987, there have been a number of impressive papers on the use of behavioral models to generate computer animation. In this approach, dubbed “behavior-based animation”, the animated creatures are endowed with the ability to decide what to do, and the ability to modify their geometry accordingly. Thus, the animation arises out of the actions of these autonomous creatures rather than from the keyframes of an animator. However, with a few exceptions, the behavioral complexity of the creatures created to date has been limited. Typically, the creatures pursue a single goal or display a single competence. For example, work has been done in locomotion [Badler93, Bruderlin89, McKenna90, Zeltzer91, Raibert91], flocking [Reynolds87], grasping [Koga94], and lifting [Badler93].

Tu and Terzopoulos' fish [Tu94] represent one of the most impressive examples of a behavior-based approach to animation. In their work they have built autonomous animated fish which incorporate a physically-based motor level, synthetic vision for sensing, and a behavior system which generates realistic fish behavior. Yet even in this work, learning is not integrated into the action-selection mechanism, the fish address only one goal at a time, and the action-selection architecture is hard-wired, reflecting the specifics of the underlying motor system and the repertoire of behaviors they wished to demonstrate.

However, consider the problem of building a digital pet such as a dog who is intended to interact with the user over extended periods of time. If the behavior of the dog is intended to be “lifelike” and if a rich interaction with the user is desired then there are a number of challenges which must be overcome. These include:

- The behavior must make sense. That is, at every instant it should choose the actions which make the most sense, given its internal state (e.g. level or hunger, thirst, or desire to play), its perception of its environment, and its repertoire of possible actions. Moreover, the temporal pattern of its behavior should make sense as well. If it is working on a given goal, it should continue working on that goal until either the goal is satisfied or something better comes along. That is, it should be able to balance persistence with opportunism and a sense of whether it is making progress, i.e. it should not get stuck in “mindless loops”.
- It should adapt its behavior based on its experience. Despite being born with a rich repertoire of innate behaviors a dog learns a great deal over the course of its lifetime. For example, it may learn explicit tricks from its owner. However, in addition, it also learns where the food dish is, what time it is typically fed, the best strategy to get its owner to open the door and to let it out, clues that indicate a walk is forthcoming, or even signs that its owner is in a bad mood and best avoided. Similarly, our virtual dog should be able to learn these types of lessons.
- It should be capable of conveying its motivational state to the viewer. Part of the fun of pets such as dogs is that we attribute clear motivational states to them on the basis of their motion, posture, direction of gaze, etc. That is, we feel as if we “know” when they are happy or sad, aggressive or submissive. Similarly, the magic of the great Disney animators lies in their ability to convey the apparent “emotions and feelings” of their characters through the quality of the character’s motion. Thus, if our virtual creatures are to be appealing to the viewer, they must provide the viewer with insight into their motivational state. Not only does this make the interaction richer, but it also helps make their behavior more explicable.
- It should be controllable. Typically, these creatures are going to be used within the context of larger system, for example, an interactive installation or story-telling environment. In these contexts, the director (human or otherwise) may want to control the creature’s behavior to a greater or lesser extent so as to achieve the larger goals of the installation or story.

Addressing these challenges will bring us squarely up against what Brooks, describes as the second research vista for Behavior-Based Artificial Intelligence in “Intelligence without Reason” [Brooks91]. That is:

> “Understanding how many behaviors can be integrated into a single robot. The primary concerns here are how independent various perceptions and behaviors can be, how much they must rely on, and interfere with each other, how a competent complete robot can be built in such a way as to accommodate all of the required individual behaviors, and to what extent apparently complex behaviors can emerge from simple reflexes”

Brooks [Brooks91] then goes on to list several issues which will need to be addressed in order to solve this problem. They include:

- **Coherence**: Given a multiplicity of goals and behaviors, how will coherence of actions and goals be insured? That is, how will the system display “just the right amount of persistence”.
- **Relevance:** How will the system take the most salient actions given its environment and internal state or motivations.
- **Learning**: How can learning be integrated into the system so it improves with experience?

To these issues we add the two additional issues mentioned above which are of particular relevance to the problem of building autonomous animated creatures:

- **Intentionality:** How can the system convey its motivational state and intentionality (i.e. what it is likely to do next) in ways that an observer will intuitively understand?
- **External Direction:** How can external control be integrated so as to allow an external entity to provide control at multiple levels of abstraction?

This thesis argues that key ideas from ethology (i.e. the study of animal behavior) may be combined with ideas from classical animation and profitably incorporated into a computational architecture for autonomous animated creatures. Ethology is a valuable source for ideas for three reasons. First, ethologists address relevant problems, namely “how are animals so organized that they are motivated to do what they ought to do at a particular time” [McFarland89]. Thus, they wrestle with the issues of relevance, coherence and adaptation within the context of animal behavior. Second, they have a bias toward simple non-cognitive explanations for behavior. They stress that seemingly intelligent behavior can be the result of very simple rules or from the interaction of what Minsky later called “many little parts, each mindless by itself” [Minsky85]. In addition they emphasize that the behavioral and perceptual mechanisms of animals have evolved to opportunistically take advantage of whatever regularities and constraints are afforded them by the environment. Third, they tend to analyze behavior at the same level which we wish to synthesize it, i.e. in terms of black boxes such as “avoid” or “chew”. Thus, they are less concerned with how these behaviors are implemented at the neural level, than with understanding what the behaviors are, and how they interact.

Classical animation is also a valuable source for ideas, because after all, classical animation is in the business of conveying motivational state and intentionality through movement. The great Disney animators are masters at this, not only in understanding how creatures convey their apparent motivational state, but also in understanding the clues which we use to deduce motivational state.

From ethology, we draw on the conceptual models developed by ethologists to explain how animals organize their behavior and adapt. Specifically, we have developed an ethologically inspired model of action-selection which by explicitly addressing the issues of relevance (i.e. “do the right things”) and coherence (i.e. “show the right amount of persistence”) improves on previous work in action-selection for multi-goal autonomous agents. In addition, we have integrated an ethologically inspired approach to learning which by incorporating temporal-difference learning into this architecture allows the agent to achieve its goals either in new ways or more efficiently and robustly. From classical animation, we draw on the "tricks" used by animators to convey the intentionality and personality of their characters. Specifically, we have developed a motor-system architecture which provides the level of coherent, concurrent motion necessary to convey the creature's motivational state (or states, as in being both hungry and fearful). In addition, we show how the entire architecture allows multiple levels of control by an external entity which may need to manipulate, in real-time, the behavior of the creature.

We have built a tool kit based on these ideas, and used it to develop a number of animated creatures including Silas, a virtual animated dog, which is one of the most sophisticated autonomous animated creatures built to date. Based on results generated from our implementation of Silas and other autonomous creatures, we will argue that this architecture solves a number of problems which have traditionally been difficult problems in the areas of action-selection and learning. While our domain is animated autonomous agents, the lessons from this work are applicable to the more general problem of autonomous multi-goal, adaptive intelligent systems.

We claim four major contributions based on the work presented in this thesis:

- An ethologically-inspired model of action-selection and learning.
- A firm (i.e. less ad-hoc) basis for the field of behavioral animation.
- The Hamsterdam tool kit for building and controlling autonomous animated creatures.
- Silas as a robust, working example.

1.1 Roadmap to Thesis

We begin our discussion of ethology and Interactive Creatures in the next chapter where we provide our motivation for pursuing this topic. Here we argue that autonomy, at some level, is an important addition to all interactive characters, and is especially important if one is interested in building "lifelike" characters. We discuss what we mean by "lifelike", and then we suggest a number of practical applications for interactive creatures as well as the underlying scientific issues that are addressed through this work. In chapter three, we summarize the five hard problems associated with building interactive creatures. and go into some depth on each one to give the reader a sense of the issues that any framework or architecture for building these creatures must address. Having laid out the issues in chapter three, in chapter four we suggest a conceptual framework for addressing these issues. Specifically, we argue that both ethology and classical animation represent fertile areas for ideas. We then suggest how each problem might be addressed from the perspective of ethology or classical animation. In chapter five we present our computational model which follows rather directly from the ideas presented in chapter four. Our presentation of the computational model is organized around the problems we are trying to solve. In chapter six we present results which demonstrate the usefulness of our approach. These results are based on our experience building creatures for the ALIVE project (ALIVE is a virtual reality system in which the participant interacts with autonomous animated 3D creatures), as well as, specific experiments which highlight particular aspects of the architecture. We conclude chapter six with a review of a number of open issues. In chapter seven we review related work, and put our work in perspective relative to other work in the areas of action-selection, learning and interactive characters. Finally, in chapter eight we reiterate the major contributions of this work.

2.0 Motivation

Do we need autonomous interactive characters? Are there practical applications for this work? Are there underlying scientific issues which have application outside the narrow area of building better and more lifelike interactive characters? In this section, we will provide the motivation for our work by addressing these important questions.

2.1 Interactivity vs. Autonomy: Is There Life After Mario?

An interactive character named "Super Mario" was one of the minor hits of the Siggraph conference this year. He is the star of Nintendo's new 3D "Super Mario" game which runs on their new Ultra-64. The game involves Mario running around in what is effectively a maze, collecting things and over-coming obstacles, all at a furious pace. The game is highly interactive as all of Mario's actions are driven solely by the user. Superb animation makes him highly engaging as well. Indeed, it is going to be a big hit at Christmas judging by the response of the conference attendees. Upon seeing Super Mario, Tom Porter from Pixar raised a very interesting question: what would autonomy bring to Super Mario? Indeed, is it necessary to have autonomous animated characters? What do we mean by autonomous anyway?

According to Maes, an autonomous agent is a software system with a set of goals which it tries to satisfy in a complex and dynamic environment. It is autonomous in the sense that it has mechanisms for sensing and interacting with its environment, and for deciding what actions to take so as to best achieve its goals[Maes93]. In the case of an autonomous animated creature, these mechanisms correspond to a set of sensors, a motor system and associated geometry, and lastly a behavior system. In our terminology, an autonomous animated creature is an animated object capable of goal-directed and time-varying behavior. If it operates in real-time and interacts with the user in real time, then it is an interactive autonomous creature.

Fundamentally, autonomy is about choices, and about being self-contained. The implicit assumption is that the agent is constantly faced with non-trivial choices, and must decide on its own how to respond. Examples of non-trivial choices include choosing between two important goals, or deciding what to do on the basis of incomplete or noisy data from its sensors. It is self-contained in the sense that it does not rely on an external entity, be it a human or a centralized decision-maker to make its decisions for it. Indeed, much of the motivation for "user interface agents" is that by taking on the job of making choices, they reduce the workload of the user.

Now Mario is highly interactive, but is not autonomous. He has no autonomous choices to make because they are made for him by the user. Indeed, part of the fun of the game is driving Mario around, and in this light you would not want Mario to be any more autonomous than you want your car to be autonomous. In addition, Mario has a very constrained world with clear choice points and very trivial choices (of course the speed of the game makes it non-trivial to play). The interaction with the user is simple, but high-bandwidth since the user is driving Mario in real time. Just as one doesn't read the paper while driving down the autobahn, one doesn't do anything else while playing "Super Mario". In short, at first blush it would seem that Mario doesn't need autonomy.

However, imagine that even if he did not have any autonomy at the level of locomotion, he had the ability to respond “emotionally” to the user’s commands. Here his autonomy is limited to choosing the most appropriate expression given the user’s actions. Thus, if the user is making choices which weren’t progressing the game, Mario could look frustrated, but look happy if the user was, in fact, making the right choices. Similarly, if Mario was at a point in the game where he was going to face some obstacles he could look worried, particularly if he had been there before and the user had screwed up. Or suppose that he could tell that the user was having problems, he could look sympathetic, or perhaps through his movements or gaze indicate the correct path or action to take. The point here is that even a small amount of autonomy could lift what is already a great interactive character into a new dimension where not only is there the fun of playing the game, but there is also the fun of seeing Mario respond on an emotional level to your direction.

As the previous discussion suggests, autonomy need not be an all or nothing thing. Rather a character can be profitably endowed with varying levels of autonomy depending on the nature of the character, the desired interaction with the user, and the nature of its environment. Below we characterize three kinds of characters each with a different level of autonomy:

- The character is a direct extension of the user, but the desired level of interaction is such that the user wishes to provide control at a high level and rely on the competence of the character to accomplish the task. The classic example of this is a web-based avatar which knows how to move about in a space without going through walls, and perhaps who “knows” the basics of interpersonal skills (e.g. knows to face another avatar when “talking to it”). This would free the user from the high bandwidth control of Super Mario, while still presenting a compelling interaction.
- The character is not directly driven by the user but interacts with the user and other characters in a relatively structured environment. A non-player character in a multiplayer game, or an adaptive opponent in a single player game, or a companion in an interactive story-telling environment would all be examples of this. More autonomy is required than in the first case because it may have non-trivial choices to make in real-time, and these choices may depend on the character’s goals and motivations as well as on its past history.
- The character is intended to give the illusion of being “alive” and of having an existence independent of the user. Examples of this include “digital pets” or virtual animals in a digital diorama. Here the interaction and environment is potentially much less structured than in the previous two cases, and the time-span of interaction is measured in weeks or months. Further, it not sufficient simply to be capable of autonomous action, the character must possess “life-like” qualities as well (see below).

Thus, varying degrees of autonomy may be useful depending on the application. But how do we endow interactive creatures with the ability to demonstrate robust, autonomous, behavior at whatever level is desired? What are the important design issues? Much of this thesis is focused on these questions. But, in addition, we are particularly interested in understanding how to build interactive characters which approach what Frank Thomas and Ollie Johnson called “The Illusion of Life” [Thomas81].

2.2 Interactive Creatures and the Illusion of Life

Life-like interactive characters are interactive characters which through their behavior create the illusion that they are, in fact, “alive”.[1] Ideally, the illusion is so complete that a user would feel badly “turning off the computer” or doing something that would “hurt” the creature. Note that being life-like or “alive” is not the same as being realistic. For example, Karl Sims’ creatures [Sims94] seem alive even though they are clearly not modeled after real animals. By comparison, Badler’s Jack [Badler93] seems anything but alive (note, this is not a criticism of Badler’s work, it simply reflects the fact that Badler is interested in other issues). Nor does it simply imply that the user uses the same social conventions to interact with it as they do with other humans. For example, Cliff Nass suggests that people treat their computers like other people. While they may do this, I would suggest that very few people attribute “life” to their computer.

But what qualities must an animated creature possess to cross the threshold from being simply an interactive character, to one to which a person attributes “life”. Below I list what I consider to be some of the main characteristics that need to be addressed in order to create this illusion.

- **The creature must react.** The end-user must “feel” as if the creature is aware of changes in its environment and react appropriately. The reaction may be as simple as a momentary shift in gaze or as elaborate as a sophisticated mating ritual. However, the creature’s awareness should be consistent with the creature’s sensory capabilities. For example, a virtual dog should not be aware of purely visual events outside of its field of view.
- **The creature must be seen as having an independent existence.** That is, the character needs to have its own agenda (i.e. goals and motivations) upon which its autonomous behavior is based. While part of that agenda should be to interact with the user, the creature should have the option not to do what the user wants if it has something more important to do.[2] It goes without saying that the creature’s independent existence needs to make sense and be interesting to the end-user since some measure of the pleasure of interacting with these creatures must come from watching them go about their way. This, after all, is an important reason why people like pets and watching animals.
- **The creature must have choices to make.** The creature’s independent existence, the environment and the interaction with the user all mean that the creature has to make choices about how to spend its time, and how to react to events in its environ-

1. While our emphasis is on interactive animated characters, the qualities discussed in this section apply to animated characters as well. That is, it is not essential to the “illusion of life” that the character interact with, or be aware of, the user. After all, Sims’ creatures create this illusion yet are not interactive. Our emphasis on interactivity comes from our belief (one shared by folks at Pixar and Disney alike) that the need for autonomous characters is greatest in interactive environments in which it would be difficult or impossible to script the entire interaction between the creature and the user.
2. In practice, care must be taken here so that the user does not simply conclude that the creature is “broken”. For example, in ALIVE, Silas will give the user a guilty smile if the user is gesturing and Silas is ignoring the gesture while performing some other behavior. Thus the creature should respond to the user, even if though it may not do what the user wants.

ment. It is a given that the creature be minimally competent at achieving its goals. Thus, issues like “do the right things” and “do them for the right amount of time” are taken as prerequisites.

- **The creature must reveal its intentionality.** The creature must have some means of conveying its intentionality. Typically, this will be through motion, the quality of that motion (e.g. slow or fast, smooth or jerky) or modification of its body (gaze, expression, posture, gesture, etc.). Note, this does not mean that the creature may always reveal its “true” intentionality (i.e. it may behave deceptively), but it does mean that the creature is conveying an intentionality which makes sense within the context of the creature’s goals[3].
- **The creature must appear to care what happens to it.** This follows from the points above but it is absolutely essential that the end-user be able to attribute “feelings” to the creature. It is this attribution of feeling which prevents most of us from harming animals: at some level we believe that the animal can feel pain and thus cares about what happens to it [Dennett96]. This may take the form of an emotional response to events in its environment. The emotional response needs to be clear to the user and needs to be consistent with the creature’s agenda. For example, it should display pleasure when it achieves a goal and anger/frustration when it is prevented from achieving a goal. The pattern of response should be believable (i.e. reflect habituation) and graduated (e.g. display varying degrees of pleasure).
- **The creature must adapt.** The creature should be able to learn from past experience so as to improve its ability to achieve its goals or to avoid unpleasant experiences. The ability to adapt needs to be consistent with the kind of creature which is being modeled. If the creature is intended to interact with the user, then it should be clear to the end-user that the character knows about the end-user and appropriately modifies his behavior based on the user’s past and current actions.
- **The creature must display variability in movement and response.** Variability and noise are characteristics of all living things: living things never do the same thing exactly the same way more than once.[4] For example, at the motor level we may not be consciously aware of noise, but we are very aware of the absence of noise which quickly leads to an impression of being robot-like. This last point is one of Ken Perlin’s most important insights [Perlin96]. Variability at the behavioral level is also extremely important: an animal’s response to a given stimulus depends on a multitude of factors including its past history and experience with the stimulus (i.e. habituation and sensitization), as well as its current motivational state. The result is that an animal rarely responds to a given stimulus in the exact same fashion every time. [5]

While other lists of necessary attributes to create the illusion of life may differ, every list is likely to contain aspects such as: the behavior needs to make sense given the many

3. Indeed, conveying deception is a hard problem because the viewer must be given clues at some point that indicate that the creature was deliberately misleading the viewer as to its real intentions rather than just being broken. In classical animation, for example, a character will often give the audience a “sly grin” before it acts deceptively toward another character.
4. This point carries over to the importance of differences between individuals, i.e. no two individuals do the exactly the same thing the same way. Chuck Jones the creator of Bugs Bunny and Wiley E. Coyote says that: “it is the individual, the oddity, the peculiarity that counts” in terms of conveying personality [Jones89]

goals of the creature, the quality of the motion needs to be life-like and reveal motivational state, the behavior should change based on past experience, etc. The computational model which I present in this thesis is intended to be an important first step toward providing these characteristics. We should note that it is questions such as "how does one juggle multiple competing goals" or "how does one dynamically convey competing motivational states" which sets the study of autonomous animated creatures apart from robotics, and makes this such a challenging area.

2.3 Interactive Autonomous Creatures: Practical Applications

There are a number of practical applications for interactive autonomous creatures:

- **As the basis for "smart avatars".** People probably can't or don't want to control avatars at the level of real-time motor control. It seems preferable to control avatars at the level of intention, and rely on the competence of the avatar to perform the task. This may demand a range of competence from simple abilities (e.g. "walk to this point" or "look like you are interested in what this person is saying") to cases in which choices may be required and you want the avatar to autonomously do the right thing (e.g. "go to the door and avoid Frank"). As the desired level of competence rises so does the need for the kind of capabilities discussed in this thesis. This will be particularly true as these worlds become more complex and as people are given tools to build their own characters or to modify the worlds on the fly.
- **As interesting non-player characters.** Whether one believes that the future of games lies in multi-person, network-based games or in single-user games, there will be a need for intelligent "non-player" characters who respond appropriately to events and other characters (including players) in the game. Once again, the desired level of behavioral complexity may span a wide spectrum from autonomous members of a crowd to a major character which you want to respond autonomously to events, and perhaps influence the outcome of the game. Indeed, non-player characters may be one of the vehicles that the game creator uses to control the flow of the game. The richer the desired behavior and interaction the greater the need for the kind of approach described in this thesis. This may be particularly true in multiplayer games which may span days or weeks and involve many players. The dynamics of the interaction may be such that it will be extremely difficult to "script" the character's behavior.
- **As smart, adaptive opponents.** Few things are as frustrating for a parent as watching your child, in tears, struggling to play a computer game against an overwhelming opponent, or floundering in the attempt to solve a difficult puzzle in an adventure game. A game and its players should adapt their play so as to be fun for the participant. This can take several forms. At the simplest level, the character should adapt its

5. Even with so-called Fixed Action Patterns one sees a good deal of flexibility. As Gould puts it: "In the face of withering criticism from psychologists through the 1950s, attention has shifted more to the flexibility of innate behavioral responses (which was noted but not emphasized in early studies). Fixed-action patterns are now more commonly called "motor programs" a designation that simply means that the behavior is generated by a specific set of neurons". [Gould94]

play to the experience level of the child, and perhaps provide clues and guidance. As the child's experience level grows, the sophistication of the opponent's play should increase as well, perhaps remembering particular things about the player's tactics and habits so it can incorporate them into its own play.

- **As companions in interactive story-telling environments.** Here the creature might be a character in an interactive story who both takes part in the story as well as leading the child through the experience. For example, the creature might learn what the person likes and dislikes and alter its behavior and maybe the flow of the story accordingly. Or the creature might be part of a story constructed by a child. Here the child would act as a director and the creature as a character. While presumably the child would be telling the character "what to do", here as in the case of avatars in general, it is extremely important that the creature be capable of a certain level of autonomous behavior so the child can interact with it at a high level (e.g. "wait under the bridge until the goat comes over the bridge and then jump out and chase him").
- **As "digital pets" and "digital creatures".** I predict that this will be an enormous area of interest in the future. Witness the fact that over 700,000 people have visited the "Dogz" website [Dogz96]. The goal here is to create digital creatures which are for all intents and purposes "alive". These creatures may be digital versions of real animals such as dogs or cats, or perhaps animals which one couldn't have as real pets because either they are extinct (e.g. dinosaur), too dangerous (e.g. lion or baboon), or fanciful (e.g. an evolved creature). The reaction to the evolved creatures of Karl Sims is telling here [Sims94]. While his creatures are very simple, the fact that they were evolved and not explicitly designed gives them a certain wondrous quality, one that seldom fails to enthrall an audience. Here the need for having life-like capabilities is obvious although it should be noted here that the goal is not necessarily realistic behavior, but rather behavior which is fun and interesting.

  A closely allied area would be in the design of creatures for "digital dioramas" for museums and zoos. These are immersive virtual environments in which the participant is surrounded by virtual creatures doing whatever they do in nature. Here the goal would be realistic and lifelike behavior. However, unlike in a zoo, or in a science museum, the participant is not just a passive observer but rather takes on the role of one of the animals and must learn how to fit in. The other creatures respond to the person as if she was one of them. The participant's actions are somehow mapped on to those of the creature.[6] For example, imagine a virtual beaver pond, in which the user must help re-build a dam, warn of approaching danger, or perhaps help take care of the young. The hope is that by providing this kind of compelling experience we can give the participant far greater insight into what life might be like as an animal.

While these are specific examples of interactive characters, there is another reason for pursuing this kind of work. Namely, as a solution to the hard control problem of Com-

6. There are a number of interesting user interface issues here. Two issues come to mind immediately. First, how are the user's actions mapped onto the actions of the animal? For example, how does a user indicate that she wants to chew on a tree trunk so as to chop it down? Second, how do you handle the mismatch in cognitive abilities between a user and the creature that she is "living inside"? For example, her repertoire of responses should be consistent with the responses a real beaver might give, given the beaver's cognitive and perceptual abilities.

puter Animation. While computer animation has made great strides in the past 10 years, it remains a very hard, and largely unsolved control problem. Consider the case of "Toy Story". Its main character, Woody, has over 700 degrees of freedom (200 for the face, and 50 for the mouth alone) which potentially must be adjusted by a human animator. At best, Pixar (a company of 150 people) was able to generate 3 minutes of finished animation a week. Perhaps equally disturbing is the fact that the animation is not re-usable in the sense that if Woody were to appear in a new movie or an interactive game, an animator would have to be intimately involved in each and every detail. This is a direct consequence of the fact that Woody does not "know" how to move or how to "behave", and must rely on an animator for every movement and behavior each and every time it is to be expressed.

The issue is not one of taking the animator out of the loop, but rather one of leveraging the efforts of the animator. Tools are needed to allow an animator to provide a character with a set of actions which the character "knows" how to perform. Subsequently, it should only be necessary to call on the character to perform that action, rather than to have to re-animate it. This is the focus of researchers such as Perlin [Perlin95]. However, it is still up to an animator to decide (perhaps via a script) when a character should perform a given set of actions. The focus of the work presented here is to further endow the character with autonomous behavior (e.g. not only does the character know how to gaze in a certain direction, but it also knows when to do so).

2.4 Interactive Autonomous Creatures: The Standpoint of Science

The ideas presented in this thesis have applicability beyond simply understanding how to build interactive animated characters. In particular, this work helps us understand how to build:

- **Multi-goal adaptive systems.** Much of the work in autonomous agent research has focused on systems which are designed to address a single goal(e.g. navigation), or if they address multiple goals, only one goal is addressed at a time. By contrast, animals must "juggle" multiple competing goals and choose, at every instant, on which goal to work and for how long. They also learn how to adopt strategies originally employed to address one goal to fulfill a completely different goal. By focusing on virtual creatures whose behavior is intended, at some level, to mirror that of real animals we must address the issues associated with building robust multi-goal adaptive systems.

  Autonomous creatures seldom do "just one thing", in contrast to most robots. The expression of even a single motivational state may entail multiple concurrent motor actions (e.g. walk and wag-tail). Indeed, multiple motivational states and their attendant motor actions may need to be expressed simultaneously. This need to provide coherent concurrency of action has major implications for the design of the behavior and motor systems of these creatures.
- **Systems which express their internal state and intentionality in intuitive ways.** Very little work has been done in understanding how autonomous systems can express their intentionality and internal state to an outside observer so as to allow the observer to understand on which goal the system is working, how it views its own progress towards that goal and what the system is likely to do next.

> By contrast, the problem of building an autonomous animated creature such as a dog which must interact with a user in a believable and dog-like manner forces one to address these issues and thus gain insight into the underlying computational issues. For example, to be a believable dog, the animated creature must have a variety of dog-like goals which it satisfies in dog-like ways. It must simultaneously perform multiple motor actions but do so in such a way that the behavior looks coherent to the user (e.g. thus a dog might wag its tail, move its ears to a submissive position, look at the user, turn to avoid an obstacle, all while walking toward a user). Finally, to be believable it must convey its intentionality both in terms of the goals on which it is working (e.g. this may be as simple as looking at its current object of interest), and perhaps its own sense of its progress toward satisfying that goal (e.g. scratching and yelping at a door because it needs to go outside).

The essential point here is that the domain of autonomous animated creatures provides a rich environment from which to draw lessons which are applicable to the general problems of autonomous, multi-goal, adaptive systems which must convey their intentionality in intuitive ways to an outside observer.

While our emphasis is primarily on understanding how to build things, our belief is that through this process we can gain understanding into the fundamental issues of action-selection and learning in animals, as well as a better understanding of our own minds. By building systems which, at some level, attempt to solve the same problems which animals must solve, we gain insight into both the problems and the possible solutions. By building "entire" creatures (albeit simple creatures) we gain insight into the interrelationship between perception, behavior, learning and action in a way that is difficult to achieve otherwise. Too often these issues are studied in isolation, despite the fact that they are highly interdependent in nature.

For example, suppose one wanted to prove or disprove that Society of Mind was a plausible theory of mind. How would one go about it? Given our understanding of the brain, it would be difficult to point to brain structures and correlate them with particular aspects of Minsky's architecture. A more fruitful approach would be to use the architecture to "build a mind" which was embodied in a creature and through that process begin to understand the strengths and weaknesses of the approach. The results from this experiment would not tell you whether our brains are constructed in such a fashion, but it would tell you the "kinds of minds" that could be constructed using Minsky's approach.

2.5 Summary

In this chapter we have argued that some level of autonomy is desirable in many, if not all, interactive characters. However, we also stressed the point that autonomy is not an all or nothing thing, but rather differing degrees of autonomy may be desired depending on the application. Our point in discussing what we saw as necessary components for creating the "illusion of life", was to stress that "life-like" means more than simply possessing autonomy. Indeed, an important characteristic of being "life-like" is the ability to convey intentionality and, possibly conflicting motivational states, through movement and the quality of that movement. Finally, we described a number of practical applications for these kinds of creatures, and discussed the broader applicability of this work.

Given that I want to build autonomous interactive animated creatures, I now want ask: what are the hard problems, and where can I go to get insight into solving them? This is the topic of the next chapter in which I will identify five hard problems, and suggest that by combining ideas from ethology (i.e. the study of animal behavior) and from classical animation, I can address these problems in a computational framework.

3.0 Problem Statement

Given that we wish to build interactive animated creatures, what are the hard problems which we must tackle? Our answer to this question is biased by our goal, namely that we wish to build interactive “lifelike” creatures such as virtual dogs, beavers, monkeys, and perhaps one day characters which are caricatures of humans. The fundamental problem for these creatures (as well as for their real world counterparts) is to “decide” what among its repertoire of possible behaviors it should perform at any given instant. This decision, in turn, is driven by what really matters to the creature, namely its needs. Therein lies the crux of the matter: given potentially competing needs, what need(s) should the creature address at any given instant and what actions should it take to satisfy those needs given its internal state and its perception of its environment? This is the fundamental problem that any creature with multiple goals and behaviors must address. Brooks’ issues of relevance, coherence, and adaptation as cited in the introduction are really nothing more than a restatement of this problem. We must also consider that these creatures are intended to be interactive, and thus, must react in real time. In addition, it is essential for a rich interaction that they be able to convey their internal state in ways that the user intuitively understands. To the extent that these creatures are used in a larger context, for example to tell a story, it is also essential that an external entity be able to “direct” the creatures to a greater or lesser extent.

There are then at least five fundamental problems which must be addressed in a successful architecture for autonomous animated creatures. These are:

- **Relevance -** How does the creature choose the most relevant set of behaviors to perform at any given instant based on its internal state and its perception of its environment?
- **Persistence and coherence -** How does the creature decide how long to persist in a given course of action, and how does it insure that its pattern of behavior over time is coherent (i.e. that competing behaviors don't work at cross-purposes)?
- **Adaptation -** How does the creature learn and integrate new strategies for satisfying its various goals and motivations?
- **Intentionality -** How does the creature convey its motivational state and intentionality[7] in ways that observers intuitively understand?

7. We mean “intentionality” in the sense proposed by Dennett. To quote Dennett: “first you decide to treat the object whose behavior is to be predicted as a rational agent; then you figure out what beliefs that agent ought to have, given its place in the world and its purpose. Then you figure out what desires it should have, on the same considerations, and finally you predict that this rational agent will act to further its goals in the light of its beliefs.” [Dennett87]. The important thing to note is that the assumption of the “goal”, “beliefs” and the causal connection between the goal and the actions (i.e. it is taking the actions so as to achieve the goal) is solely in the mind of the observer. Thus, for Dennett, an animated character can be seen as having intentionality, even though it is no more than a temporal sequence of images. From an animator’s perspective, their task is to make it easy for the observer to attribute intentionality to a character. This means making the goals and beliefs of the character readily apparent to the observer, and making sure that the resulting behavior is easily explained given those goals and beliefs.

- Integration of External Control - How does one integrate external control at multiple levels of abstraction so as to allow an external entity such as a director to influence the behavior of the creature?

These problems are discussed in more detail below. However, before leaving this list, two points are worth noting. First, we take as a given the need for real-time response. Second, this is not intended as an exhaustive list. There are a host of other problems which need to be considered. For example, how does the creature sense, and "make sense of" its world, including perhaps, understanding the intentionality of other autonomous creatures? Or, as Maes [Maes94] suggests, how does the system insure that it degrades gracefully in the presence of failure or unexpected problems? Or, how does the creature generate life-like motion? Or, what tools are required for an animator to build such creatures? If the creature is intended to model a human, then there are the very important issues of language and gesture understanding and production. These are all non-trivial issues, to say the least. Nonetheless, we believe that our list of five issues are fundamental ones for building any sort of interactive, autonomous, animated creature with multiple goals.

3.1 Relevance and Attention

A creature with multiple competing goals must, at every time step, pick the most relevant set of goal-achieving behaviors to perform given the creature's internal state (i.e. goals), the creature's perceived environment and its previous history. In a distributed system of competing behaviors (where a behavior is a self-interested goal-directed entity), this means that individual behaviors must evaluate their own relevance and compete for control on this basis. Thus, the fundamental problem for a behavior is to weigh the mix of relevant internal and external factors so as to arrive at a "value" with which it may compete. Implicit in this process of transducing a value, is the question of how external and internal factors should be represented and combined. Most previous systems, however, have treated internal and external factors differently, or have minimized the importance of one component or another. For example, traditional planners tended to de-emphasize external factors, whereas reactive systems have minimized the importance of goals and internal state in modulating behavior. This has made it difficult to demonstrate opportunistic behavior, or to demonstrate natural behavior in which varying levels of external and internal factors produce the same result. One of the important, albeit simple, lessons from ethology is the need to treat internal and external factors as first class objects, i.e. to represent their respective values using a common currency.

Implicit, too, in the issue of relevance is the problem of attention. That is, how does the creature decide to which features of its environment to attend in order to assess the relative relevance of its behaviors?

3.2 The Need for Coherence of Action and Goal

A creature only has limited resources to apply to satisfying its goals (e.g. it can only walk in one direction at a time), and thus there needs to be some mechanism to arbitrate among the competing behaviors. Moreover, once a creature is committed to satisfying a goal, it makes sense for it to continue pursuing that goal unless something significantly more important comes along. This need for coherence of action and goal introduces two sub-problems:

- **Temporal control:** The first problem is providing just the right amount of persistence. Too little persistence and the creature will dither between behaviors which address competing goals. Too much persistence and the creature will not take advantage of unexpected opportunities. Or it may mindlessly pursue a given goal to the detriment of other goals.This has been a difficult problem for previous computational models of action-selection in large measure because they provide no explicit mechanisms for controlling persistence.
- **Concurrency and compromise:** Pursuit of a single goal should not preclude the expression of compatible behaviors which may address other goals. Two behaviors may be compatible if, for example, they require orthogonal motor actions (e.g. “walk” and “wag tail”). Alternatively, they may be compatible because it is possible to blend their preferences (e.g. “move-to” and “avoid”) so as to arrive at a compromise solution. Or they may be compatible because one behavior seeks to modulate how another performs its action (e.g. “move-forward” and “bound”). Thus, the system must be able to support concurrency, blending and modulation of motor actions in such a way that the creature still behaves coherently. Previous approaches have either ignored this problem [Maes90a, Tu94] or have required very careful design of the behavior system [Brooks86, Tyrrell93]. By contrast, we provide a motor system architecture which explicitly supports this functionality.

3.3 Conveying Motivational State and Intentionality

If a rich and satisfying interaction is desired with these autonomous animated creatures then it is extremely important that they convey their internal motivational state and intentionality in ways that an observer will intuitively understand. Dogs or cats make good pets for just this reason, i.e. that it is easy for us to make up “stories” as to why the animal is behaving as it is. Similarly, it was this emphasis on conveying the internal state of their characters which separated early Disney animation from its competitors [Thomas81]. In both animals as well as animated characters, motivational state and intentionality is conveyed through the creature’s actions, features, and the quality of its motion. Thus, the expression of even a single motivational state may require multiple concurrent motor actions. Moreover, multiple motivational states may need to be expressed simultaneously (e.g. hunger and fear). In short the need to be able to express motivational state and intentionality reinforces the importance of concurrency and compromise at the behavior and motor level.

This should not be interpreted as meaning that the creature’s actions are always predictable, or that one always knows what is “going on” inside the creature. Indeed, as Peter Todd once pointed out, the most satisfying interaction may be one in which you don’t know exactly what the creature is going to do, but once they do it, their behavior make sense in retrospect.

3.4 Learning and Adaptation

An autonomous creature should be able to adapt its behavior based on its experience so as to achieve its goals more efficiently or more robustly. This adaptation may take the form of learning which behaviors are more effective than others in achieving their underlying objective. Alternatively, it may take the form of learning how to re-use existing behaviors in new contexts so as to satisfy previously un-associated goals (e.g. learning that “sitting” in response to hearing “SIT” leads to a reward sufficiently often that it is a useful strategy for gaining that reward). Still another form of adaptation is learning new contexts which predict the occurrence of a previously un-associated context and thereby allows a goal to be satisfied more quickly or more robustly (e.g. learning that a tone predicts the appearance of food). In each case the task facing the creature is not only to learn the lesson at hand, but also to integrate it into its existing behavior system. Previous work in learning, while impressive, has typically focused on learning optimal strategies for single-goal systems. Indeed, the work of Maes [Maes90] and Humphrys [Humpfrys96] stand out as some of the few efforts to show how learning could be integrated into an action-selection algorithm. One of the major contributions of the model presented in this thesis is to show how temporal difference learning may be integrated into an ethologically inspired action-selection architecture.

3.5 Integration of external control

One of the practical problems associated with autonomous animated creatures is the need to accommodate external control so as to allow an external entity to influence the behavior of the creature to a greater or lesser extent. For example, imagine one was designing an immersive “Lassie” experience for children. Suppose the virtual Lassie did fine job as an autonomous dog, but for whatever reason was ignoring the child. Or suppose, you wanted the child to focus on some aspect of the environment which was important to the story, but Lassie was distracting her. In either case, you would want to provide real time external control to the autonomous Lassie. For example, by increasing Lassie’s motivation to play, she would be more likely to engage in play. Alternatively, Lassie might be told to “go over to that tree and lie down” so as to be less distracting.

This example illustrates both the need to be able to integrate external control as well as the need to be able to provide control at multiple levels of abstraction. The most concrete level of control is at the motor level (e.g. lie-down). This is appropriate for tasks which do not require sequences of actions. The next higher level of abstraction is at the task level (e.g. find the user and beg for food). This frees the director from having to specify the sequence of actions required to perform the task. Note: this is the level that one might wish to control an avatar. If the director simply wants to predispose the creature to interact in a certain way, but she does not want to specify a particular task, then she can control the character at the motivation level. For example, she might increase the creature’s level of hunger to bias it to look for food. Finally, or perhaps orthogonally the director can affect the creature’s actions by altering either the environment or the creature’s perception of its environment. The point here is that depending on the application, one might need to control the creature at any of these levels of abstraction.

3.6 Summary

In this chapter we have described five fundamental problems facing an architecture for interactive autonomous creatures: relevance, coherence, adaptation, intentionality, and integration of external control. In the next chapter we turn to a discussion of the insights that may be gained by looking at these problems from the perspective of two different disciplines: ethology and classical animation. We will begin by briefly summarizing why each field offers such a valuable perspective on the general problem of building interactive creatures. Then we will look at each problem in turn from the perspective of either ethology or classical animation and focus on the key insights which may be directly incorporated into a computational model.

4.0 Lessons from Ethology and Classical Animation

In this chapter, we summarize the key insights from ethology and classical animation which underlie the computational model presented in the next chapter. However, before we begin it is perhaps worthwhile to ask the question: "Why look to ethology and classical animation for insights? That is, why should these disciplines have anything useful to say about building interactive autonomous creatures?" We begin this chapter by summarizing why these two disciplines hold great potential as sources of ideas and insights for us. We then detail what we see to be the key insights on each of the five big problems.

4.1 Why Look at ethology?

Ethology is the study of how animals behave and adapt in their natural environment. In particular, ethologists are focused on the "why" of behavior and perhaps Tinbergen's greatest intellectual contribution was to note the "why of behavior" could be addressed on four different levels[Krebs93]:

- Proximate causation: What were external and internal factors which triggered the behavior (e.g. Bruce jumped because he heard a rustling in the grass, and he had seen a snake earlier on his walk)?
- Function: What is the adaptive value of the behavior (e.g. reacting quickly may prevent Bruce from dying from a poisonous snake bite, and thus he will have a greater chance to have offspring to carry on his genes and/or increase the probability that his offspring will grow up to reproduce)?
- Development: What are the developmental factors which led to the behavior (e.g. Bruce's mother was terrified of snakes, and at an early age Bruce watched his mother scream in response to stepping on a snake)?
- Evolution: What are the evolutionary factors which led to the behavior (e.g. Bruce reacted to the snake because his primate ancestors lived in an environment populated by a zillion different kinds of poisonous snakes)?

These 4 questions have framed much of the research in ethology. For our purposes we are most interested in work which focuses on proximate causation and development. With respect to causation, McFarland puts it best when he asks: "How are animals so organized that they are motivated to do what they ought to do at a particular time?" [McFarland89]. With respect to development, the comparable question might be: "How are animals so organized that they learn what they ought to learn at a particular time?". Our hope is that the answers suggested by ethologists to these questions are directly applicable to building "interactive creatures" who are so organized that they are motivated to do what they ought to do, or learn what they ought to learn at a particular time.

The typical way that ethologists have approached these problems is to: observe, experiment and explain. Indeed, one of the reasons that ethology is such a fertile area for ideas is that there are 75 years of field observations and analysis upon which to draw. For example, the complete behavioral repertoire of certain animals such as hamsters has been carefully catalogued into so-called "ethograms". Detailed experiments, motivated by observed patterns of behavior, have been performed to elucidate exactly what it is in the environment which apparently triggers a certain behavior, or how varying levels of internal influences affect the resulting response of the animal.

However, it is in the explanations and models of ethologists that we find the greatest value. This arises from three characteristics of the models proposed by ethologists: first, the emphasis on goal directed behavior, second, their bias toward simple, non-cognitive explanations for behavior, and third, their choice of a level of abstraction at which to analyze behavior. This is discussed in more detail below.

The first characteristic is the emphasis on observed behavior as the animal's attempt to satisfy its competing physiological needs in an uncertain environment. In other words, goals are an implicit part of most Ethological models, and one of the key questions ethologists attempt to answer is how animals organize their behavior so as to satisfy these competing goals. For example, given the competing needs to drink, eat, or mate, how does the creature "decide" what to do, how long does the creature persist in addressing a given need, how does it take advantage of un-expected opportunities, and given a particular goal how does it "decide" what course of action to take? These questions are clearly very central to the problem of building autonomous animated creatures.

Second, ethologists have a tradition of looking for simple, non-cognitive explanations for behavior. Faced with the remarkable abilities of creatures such as bees or rats to behave "intelligently" with respect to the problems they need to solve in order to survive and reproduce, ethologists recognized early on that seemingly intelligent behavior could be the result of very simple rules or from the interaction of what Minsky later called "many little parts, each mindless by itself" [Minsky85]. This idea was coupled with the idea that perceptual and behavioral mechanisms evolved within the context of a specific environment and the set of problems the creature faces, and so the resulting mechanisms opportunistically took advantage of whatever regularities and constraints were afforded them by the environment. Thus, ethologists were motivated to ask: what are these simple behaviors, how do they interact, and what are the perceptual "hacks" that animals use to make sense of their world?

These ideas, of course, have had a profound effect on the field of Artificial Intelligence, although one is occasionally left with the feeling that some AI researchers took the view from 30,000 feet and missed some of the most important lessons. It was, and is, a seductive idea, fueled by books such as Braitenberg's "Vehicles" [Braitenberg84] which many people read to suggest that intelligent behavior was no more than hooking up the right sensors to the right motor actions.[8] In practice though, AI researchers soon discovered that it wasn't obvious to determine the list of individually simple behaviors or how to identify the relevant set of features to which to attend, or figure out how these behaviors all interact so as to produce coherent, intentional behavior, or what to learn and when to learn it.

Fortunately, these are the very questions which ethologists, having accepted the basic premise of this approach, have been struggling with for 50 years. This is why it is worth taking a closer look at the Ethological literature. In some cases, we can borrow ideas as

8. I doubt very much that this was Braitenberg's intention, and indeed his discussion of these simple vehicles occupies only a small part of his book.

to how behaviors might be organized so as to deal with the issues of relevance and coherence. Indeed, our computational model for action-selection is taken directly from the ideas of ethologists such as Tinbergen, Lorenz, McFarland and Ludlow. In other cases, we can borrow ideas on how the behaviors of animals is organized to make it easier to learn the things they need to learn. Once again our model is very much inspired by ideas from ethology. In still other cases, we can borrow specific hacks that certain animals use to solve particular problems. For example, our approach to using motion-energy is directly inspired by work on bee navigation and obstacle avoidance which suggests that they rely on perceived motion energy to accomplish these tasks.

The third important characteristic of the models developed by ethologists is that they have chosen a relatively high level of abstraction when analyzing and explaining behavior. Indeed, it is at the level at which we want to synthesize behavior. The models proposed by ethologists are typically conceptual and derived from observations of animal behavior rather than from brain studies. The models are intended as useful ways to think about behavior and to explain observed behavior rather than to guide research into the workings of the brain. Thus, while they may speak of structures such as behaviors, releasing mechanisms and motivational variables, these are simply abstractions to help understand the observed behavior. For example, a “releasing mechanism” is an abstraction for the set of neural processes which signal the presence of a given stimulus, and “internal motivations” are abstractions for the multitude of internal factors which influence behavior (e.g. hormones, body temperature, blood sugar levels and so on). There is no suggestion that one could point to a particular part of the brain and say “here is the avoid behavior”, or “here is where hunger flows into the system” or “here is the snake releasing mechanism” (although in the case of certain releasing mechanisms, one can point to specific feature detectors in the brain and show how they could act to identify relevant stimuli). [9]

Minsky, of course, takes a similar approach in his landmark book “Society of Mind” [Minsky88] in that he proposes an extremely powerful conceptual model for how our “minds” might be organized. He is much less concerned with how “agents”, for example, might actually be implemented at the level of neurons other than to convince one that agents simple enough to be plausibly implemented via networks of neurons could collectively produce complex behavior. This is the right level for us to work as well given our focus on building interactive characters with rich and robust behavior. What we care about is the observed behavior over time and not necessarily the biological plausibility of the actual implementation (to paraphrase Papert “biology should be on tap, not on top”). Thus, while we will speak of behaviors, releasing mechanisms, motivational variables, and behavior groups “as if” they were real things, please understand that we are keenly aware that they are merely useful abstractions for the result of complex biological processes which we only partially understand.

9. As Gould [Gould82] points out, one of the early stumbling blocks of ethology was to explain how innate behavior (and in particular innate releasing mechanisms) could be transmitted via genes. For example, how could the image of a mother herring-gull be encoded in the genes of a baby herring-gull which at birth seemingly recognizes its mother. The answer, undoubtedly to the great relief of the ethologists, was that the herring-gull baby was responding to a vertically moving red dot, and subsequent brain studies demonstrated the existence of neurons in the visual cortex of herring gulls which fired in the presence of such stimuli.

Another useful abstraction which permeates many Ethological models is the notion that behaviors compute a number which represents some measure of the “value” of the behavior to the creature (e.g. the contribution of the behavior to the creature’s fitness) and the behaviors compete on the basis of that “value”. It can be catching, even a noted neurobiologist such as Joseph LeDoux started his classic paper on the neural basis for emotion by stating: “Brains compute value...Intervening between the transduction of raw stimulus energy and the initiation of protective motor responses are central processes which determine the value of the stimulus for the individual organism. On the basis of computed value, the brain decides whether a response is required and what the response should be.” [LeDoux90]. McFarland, for example, argues that the tools of economics can be used to help explain animal and robot behavior. [McFarland93b] Similarly, behavioral ecologists have shown that in certain cases the observed behavior of animals is a near optimal response if the animals were attempting to maximize or minimize some entity (such as net energy consumption)[10]. The point here is that the idea of behaviors being able to compute a number which represents some measure of the “value” of the behavior is a useful and pervasive abstraction, and one which we will use here. Nonetheless it is important to remind ourselves from time to time that this is an abstraction, and that at best all we can say is that the animal is behaving “as if” the behaviors were competing on the basis of some computed value.

It should be noted that our approach to learning is informed by the practice of animal training[Lorenz94, Rogerson92]. That is, by understanding the techniques used by animal trainers we hope to better understand the underlying mechanisms which constrain the learning process in this particular instance.

Thus, for all of these reasons, ethology represents a fertile place to look for answers. However, before we describe the Ethological perspective on the problems presented earlier, we will briefly outline why classical animation is also an important source for ideas.

4.2 Why Look at Classical Animation?

Given our objective, it is hardly surprising that we also look to classical animation for inspiration. Like ethologists, animators are concerned with observable behavior, but in the case of animators they are interested in how observable behavior conveys motivational state and intentionality in ways that we intuitively understand. As Frank Thomas of Disney put it: “We struggled to build interesting, appealing characters, but most of all we worked to find ways to make the audience feel the emotions of the animated figures -- emotions the audience could relate to, identify with, and become involved in” [Thomas81].

The fundamental insight of the Disney animators was to understand that motivational state and intentionality could be conveyed through the motion of the character and

10. My favorite example of this is a kind of crow which picks up shellfish, climbs to a certain height depending on the weight of the shellfish and then drops them so as to break open the shells (a process which may need to be repeated several times depending on the size of the shell). It appears that these crows are “programmed” to minimize the total height required to break the shells[Krebs93].

through the quality of that motion. In other words, its not just what you do, but how you do it that counts. In addition, the magic of the great animators was that they "knew" the set of clues that you had to get right in order to create the "illusion of life". They knew that in a turn if the eyes didn't lead the head, and if the head didn't lead the body, the resulting motion would not look right. But they also knew that it wasn't necessary to be "realistic" to be "believable and life-like". After all, Pluto is not a realistic dog by any stretch of the imagination, but he is a believable dog because the animators got the small set of features that comprise "doginess" right.

Along these lines, Tom Porter of Pixar tells an interesting story. They had some of their animators come in to advise them on "walk cycles" for an interactive character on which they were working. The first question the animators asked was: "But what is the character feeling?" In the minds of the animators, there was no such thing as a generic walk. The state of the character and the quality of the walk were inextricably intertwined. [Porter96]

If from ethology we borrow ideas on how interactive animated creatures should organize their behaviors, and from animal training on how they adapt, then from classical animation we borrow ideas on how they should convey their motivational state and intentionality. In particular, we are interested in understanding the implications of the techniques of classical animators for the design of the creature's motor and behavior systems.

We now turn to a summary of what we take to be the key insights from these two fields on the five big issues described earlier. In the next chapter we will outline our computational framework which incorporates these ideas.

4.3 Ethological Perspective on Relevance and Attention

Ethologists from Tinbergen on have suggested that it is useful to view the observed behavior of an animal as the outcome of competition among competing behaviors, where each behavior is a self-interested goal directed entity [Tinbergen50, Lorenz73, Baerends76, McFarland93, Dawkins76, Gallistel80]. Behaviors are viewed as being organized into collections or groups each of which addresses a specific biological function such as feeding or mating. Behaviors compete for control of the creature based on their relevance given the creature's internal state and perceived environment.

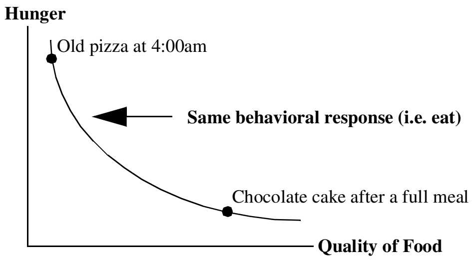
Ethologists have long stressed the importance of internal factors in determining the observed behavior of animals. That is, they viewed the behavior of animals as a response to perceived changes in their state. Certain systems, such as feeding, were viewed as "homeostatic" in that animals were viewed as behaving so as to keep certain internal factors within some delta of a set-point. The greater the difference between the actual value of the internal factor and its ideal set-point, the greater the tendency of the animal to behave so as to reduce the difference. For example, Lorenz proposed the so-called "hydraulic model" in which "action-specific energy" would accumulate, which in turn would lead to behavior which would result in a stimulus situation which would trigger a sensory filter (called a Releasing Mechanism) which in turn would release a behavior which would reduce the "action-specific energy" [Lorenz73, McFarland93]. Similarly in Tinbergen's hierarchy internal factors largely direct behavior, particularly in the upper levels of the hierarchy[Tinbergen53]. While few researchers today would argue that Lorenz's model is correct in its details, the importance of internal factors in helping to determine behavior is well accepted.

McFarland, while not denying the importance of internal factors in determining behavior, has argued that the ideas of the early ethologists were gross simplifications, i.e. "the interactions among the physiological processes that give rise to behavior are complex" [McFarland93]. For example, "hunger" is not a one dimensional variable. That is, the animal may behave differently depending on whether it has a salt or thiamine deficiency. Similarly, behavior rarely affects only a single internal factor. For example, drinking water reduces thirst but also changes the body temperature. The fact that we and many animals eat at particular times of the day suggests that more than simple homeostasis may be required to explain our feeding patterns. Indeed, as we will discuss below different combinations of internal and external factors can produce identical behavior.

Once again, McFarland's point is not to suggest that internal factors aren't important, but rather to recognize that behavior is the result of a potentially complex interplay of external and internal factors. One also needs to recognize that while it is convenient to speak of motivational variables such as "hunger" or "thirst", these are really only coarse grained approximations to the "sea" of internal influences on behavior.

Each behavior is viewed as being responsible for assessing its own relevance. It does this by weighing the mix of relevant internal and external factors so as to arrive at a "value" with which it may compete with other behaviors for control of the creature. It is well known that different combinations of internal readiness and external stimuli can result in the same behavioral response (e.g. eating chocolate after a full meal and eating a slice of pizza of dubious origin at 3 in the morning). This phenomena is illustrated in Figure 1 ethologists argue that this can be explained if behaviors evaluate internal needs

Figure 1 Differing levels of internal and external factors can produce the same behavioral response.



Internal and external factors need to be expressed using a common currency if differing levels of external and internal factors are to produce the same behavioral response. This is a common phenomena found in nature. The actual shape of the curve is a function of the method used to combine the strengths of the relevant factors, and in animals it is ultimately determined by the animal's needs and the characteristics of the its environment.

and external opportunities using a common currency [Lorenz73, Baerends76 and McFarland93]. This implies 2 key points. First, some mechanism associated with the behavior needs to transduce a value from external stimuli, where the value is expressed in the same units used to represent the value of internal factors (e.g. a continuous quantity). Second, the behavior needs to be able to combine the values of the relevant factors using some operation such as addition, multiplication, or some mixed rule.

Opportunistic behavior arises out of the relative weight placed on external factors versus internal factors. For example, for a desert animal, the mechanism which transduces a value from stimuli associated with water may output a higher value than the mechanism which signals the presence of food.

Implicit in the problem of relevance is the issue of how a creature decides on the set of features in their environment to which to attend? Tinbergen and Lorenz [Tinbergen53, Lorenz73] were among the first researchers to posit the existence of what they called Innate Releasing Mechanisms (or simply Releasing Mechanisms). A Releasing Mechanism is an abstraction for the collection of feature detectors, or perceptual routines which identify a particular object or event of biological significance. Releasing Mechanisms are conceptualized to be simple, fast, and just adequate [Gould82]. For example, an Ethologist would say that people have releasing mechanisms for identifying the presence of snakes, and this is why we (or at least the author) jumps when we see a coiled rope or hear rustling in the grass. In another example, baby herring gulls have a releasing mechanism for identifying their mothers when they hatch. It is basically a feature detector which fires when it sees a red dot moving up and down. This works because herring gulls have a red dot on their beaks. Thus, baby herring gulls are easily fooled by a moving popsicle stick with a red dot on one end. [Gould82] On the other hand, save in the presence of an overly enthusiastic student of ethology, this is a rare occurrence in nature.

Interestingly enough, Ramachandran [Ramachandran90], while not an ethologist, argues for a similar idea in the early stages of visual processing. That is, "perception of apparent motion is controlled by what is in effect a bag of tricks, one the human visual system has acquired through natural selection during millions of years of evolution... In the real world anything that moves is a potential predator or prey".

A Releasing Mechanism represents an example of a partial representation of an aspect of the world. It is tied to a specific behavior or group of behaviors, and is only sensitive to whatever set of features are minimally necessary to identify a relevant object or event for the purposes of the behavior. The discrete and task-based nature of releasing mechanisms means that different releasing mechanisms may be sensitive to different features of the same object. Yet there need not be a coherent and complete representation of the object, because it may not be necessary.

4.4 Ethological Perspective on Coherence

Ethologists have always been particularly interested in the temporal pattern of animal behavior, perhaps because animals do such a remarkable job at showing the "right amount of persistence" and coordination of movement. Ethologists generally agree that animals primarily address one goal at a time [Dawkins73, Tinbergen50, Lorenz73, McFarland75, McFarland93, McCleery83] without dithering among behaviors which work at cross-purposes and tend to persist in performing goal-compatible behaviors until the goal is satisfied. However, given an unexpected opportunity to satisfy another goal, they will often interrupt the pursuit of a current goal to take advantage of this opportunity. In fact, sometimes animals appear to engage in a form of time-sharing [McFarland75, McFarland93], in which low priority behaviors are given a chance to execute, despite the presence of a higher priority behavior. In other cases animals act as if they "know" the expected time course associated with a behavior (e.g. how long to engage in a behavior before a reward is presented) or whether they are making the expected level of progress, and will adjust their persistence accordingly [Gallistel90,94]. All of these are typically hard problems for autonomous creatures.

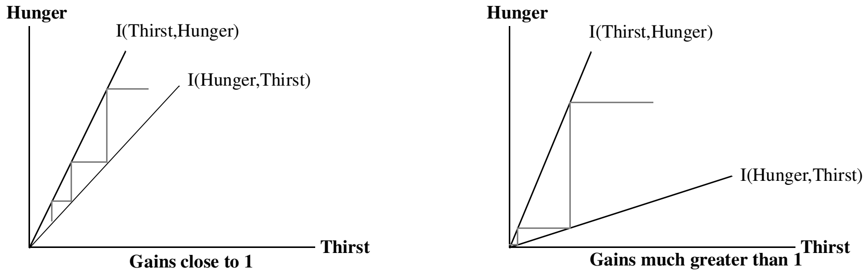
On a different level, although we take it for granted, animals consistently produce coordinated motion of their multiple degree of freedom bodies. For animated characters this too is a non-trivial control problem.

These capabilities seem to imply a number of things. First, behaviors are constantly competing for control. Behaviors may compete if they require control of the same underlying motor resources (e.g. a limb or a joint), or if they represent different strategies for addressing the same functional need. This suggests that the behavior system must be organized into groups of competing behaviors and there must be a robust arbitration mechanism. Second, the need for persistence suggests that the arbitration mechanism needs to be such that an active behavior has the equivalent of inertia which makes it likely to remain active. However, there need to be a variety of mechanisms in place to insure that the persistence or inertia is not so high so as to preclude taking advantage of unexpected opportunities or persisting past the point of reason. Third, the underlying motor system needs to be capable of producing coherent concurrent motion.

As mentioned in the section on relevance, ethologists typically view the behaviors of a creature as being organized into functional groups which serve different needs (e.g. feeding or mating). A given functional group is typically composed of behaviors which represent competing strategies. For example, a group associated with reducing hunger might have one behavior which is actually responsible for eating, and then several behaviors which represent alternative strategies for getting the creature into a context in which the eating behavior can become active (e.g. rattling the food dish, barking, or knocking over the trash). Minsky [Minsky88] calls these groups of competing behaviors (i.e. behaviors which work at cross-purposes) cross-exclusion groups. In either case, mutual or lateral inhibition is put forth as an arbitration mechanism. Both Ludlow [Ludlow76,80] and Minsky have stressed the point that via the "avalanche effect" mutual inhibition provides a robust arbitration mechanism and one which displays the inertia property described above (see Figure 2). Ludlow, in particular, noted that by adjusting the inhibitory gains, a given behavior could be made more or less persistent relative to the behaviors with which it competes. In general, the greater the gain, the more persistent the Behavior. This is shown in Figure 2

Ludlow [Ludlow76,80] also stressed the importance of "boredom", "fatigue" or some other measure of "level of interest" in modulating the persistence associated with a given behavior regardless of the intrinsic value of the behavior. Ludlow's idea was that the level of interest associated with a behavior decreased over time as the behavior was active, thus eventually allowing other less important behaviors to become more active. More generally, level of interest represents the behavior's view of how well it is doing toward achieving its goal. Its level of interest might decline purely as a function of the time the behavior has been active, or perhaps because the world isn't progressing in the

Figure 2 Mutual Inhibition provides control over persistence



Inhibitory gains can be viewed as the slopes of switching lines in motivational state-space. The dotted lines above correspond to the trajectory in state-space. Horizontal portions of the trajectory correspond to periods when the creature is addressing its Thirst, vertical portions correspond to addressing Hunger. Switching lines represent boundaries in the state-space where the creature will switch from addressing one motivation to addressing another. If the slopes of the switching lines are close to 1.0, a relatively small change in the underlying motivation will cause the system to switch behaviors (i.e. dithering is likely). If the slopes are greater than 1.0, then a much larger change in the underlying motivation will be necessary. Hence, the creature will display greater persistence, and make more rapid progress toward satisfying its goals. This assume, of course that there is a cost associated with switching behaviors.

expected manner (e.g. perhaps the behavior relies on a malfunctioning part of the motor system). This latter idea is one that Payton has incorporated into his action-selection algorithm [Payton92].

Level Of Interest also has a clear meaning in the context of an appetitive behavior: it controls how long the creature should persist in the behavior before giving up. For example, if the creature has learned from experience that a reward usually follows within X ticks of commencing some behavior, then it should base the amount of time it will pursue this behavior on this knowledge (i.e., it should accordingly adjust the amount by with the Level Of Interest is decreased after each performance of the behavior).

4.5 Classical Animation's Perspective on Conveying Motivational State

As observers, we base our explanation of an animal's (or animated character's) motivational state and intentionality largely on the motor actions of the creature and our knowledge of the creature's world. For example, we say a dog is "fearful" when its tail is between its legs, its ears are flat on its head, its mouth is closed, it slinks around with its head and neck low, and it avoids direct eye contact. Animators are masters of utilizing these "clues" to convey the motivational state and intentionality of their characters [Thomas81, Lassiter87]. Specifically, they utilize clues such as:

- Quality of Motion (e.g. bouncy walk vs. a shuffle)
- Expression (e.g. smile vs. frown, eyebrow and ear position)
- Posture (e.g. erect vs. submissive slouch)
- Focus of Attention (e.g. gaze)

As mentioned earlier even the display of a single motivational state may require multiple concurrent but coherent motor actions, and often multiple motivational states may need to be conveyed simultaneously. To model this, either the "winning" behavior has to be responsible for taking all these factors into account, or there has to be some mechanism which allows multiple behaviors to express their preferences for motor actions in some prioritized manner, and in such a way that the creature still moves coherently.

This need for coherent, concurrent motion is also lurking in Ethological models as well. While ethologists speak of animals being engaged in a single activity at a time [McFarland75] they only mean this to be true at a certain level of abstraction. For example, a dog approaching a food dish may be said to be "engaged in feeding", yet common sense tells us that (a) the dog will be performing multiple motor actions concurrently, and (b) the manner in which the dog approaches the dish (i.e. what set of motor actions are actually expressed) will be modulated by a number of factors including its level of hunger, its mood, the specifics of its immediate surroundings (e.g. the cat that is on the counter, or the 2 year-old who is bearing down on the dog from the left), and so on.

For a number of reasons, it is highly desirable that some mechanism be available to allow the "losers" in the competition for control to express their preferences for motor actions. While ethologists generally view behavior systems as being organized as loosely layered heterarchies [Dawkins76, Baerends76, Davey89, Hogan88, McFarland75] or as Minsky puts it: "... one that is riddled with short-cuts, cross-connections and exceptions" [Minsky87], any sort of hierarchy, particularly a strict winner-take-all hierarchy, does introduce a number of problems as pointed out by Maes and Tyrrell [Maes90a, Tyrrell93]. The most relevant problem for the purposes of this discussion is that in a strict winner-take-all hierarchy a decision is made at every branch point, and therein lies the difficulty. Even if the right decision is made, there is a loss of information because the losers may have preferences for motor actions which are either orthogonal (e.g. "walk" and "wag-tail") or compatible (e.g. "move-forward" and "gallop") or which when blended with the preferences of the winner would result in a compromise solution that helps move the creature toward multiple goals simultaneously (e.g. allowing "avoid" and "moveto" to combine their preferences for actions so as to allow the creature to not only make progress toward a goal, but also avoid an obstacle in its path). There is also the practical problem that is illustrated by the example at the beginning of the section. In a strict winner-take-all hierarchy the "move-to-food" behavior would have to know about all of the other subsidiary actions that it should be performing (e.g. wag tail, growl etc.). For all of these reasons, it is critical that winners and losers be able to express their preferences for motor actions while respecting the relative importance of the requests.

4.6 Ethological Perspective on Learning

Once again our approach to learning is inspired by decades of ethological research [Dickinson94, Gallistel90, Gallistel94, Gould94, Lorenz73, Sutton90, MacFarland93, MacFarland95, Plotkin93, Shettleworth94, Davey89, Toates95]. Below are some key lessons from ethology which inspire our architecture.

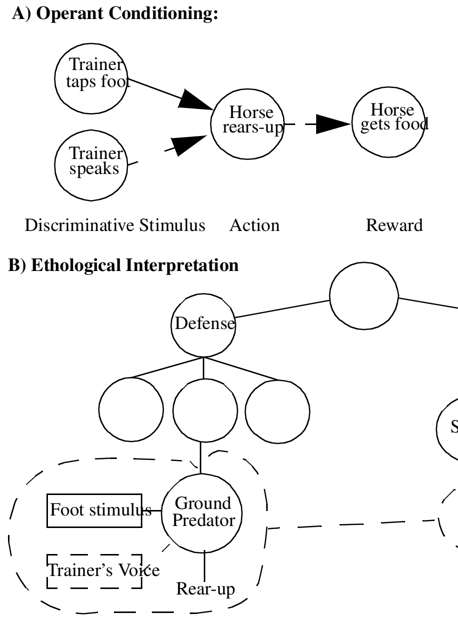
- **Learning complements evolution:** Learning is a mechanism for adapting to significant spatial and temporal aspects of an animal's environment that vary predictably, but at a rate faster than that to which evolution can adjust (e.g. predictable changes occurring within a generation). The most adaptive course in this case is to evolve mechanisms that facilitate learning of these rapidly-varying features. Thus, evolution determines much of what can be learned and the manner in which it is learned. While this is sometimes viewed as imposing negative “constraints on learning,” the innate structure in which learning occurs most often has the effect of dramatically simplifying the learning process [Gallistel90, Gallistel94, Gould94, Lorenz73, Miller90, MacFarland93, Plotkin93, Shettleworth94, Todd91].

- **Motivations and goals drive learning:** Animals learn things that facilitate the achievement of biologically significant goals. Thus, motivations for those goals drive learning. It is far easier to train a hungry animal using food as a reward than a satiated one [Gould94, Lorenz73, MacFarland93]. The components of the lesson to be learned--that is, what things or actions facilitate a particular goal--are usually, but not always, contiguous in some sense with the satisfaction of the goal. For instance when a dog gets a cookie, the thing to learn is what behavior it performed just before getting that cookie. It should be noted that temporal or spatial contiguity is only part of the story. For example, a rat will associate the smell of a food with getting sick hours after eating the food and will subsequently avoid that food. This is known as taste-aversion learning.
- **Animals learn new appetitive behaviors:** Operant or instrumental learning can be viewed as an animal's discovery of new appetitive behaviors--that is, new behaviors that bring them closer to attaining some goal [Lorenz73]. In more detail, operant conditioning occurs when an animal finds that an existing behavior (one that was typically performed in another context for another purpose) performed in a new stimulus situation will reliably lead to a state of the world where one or more motivational variables will be satisfied when the animal performs a consummatory behavior. For example, a dog that learns to roll over on command in order to get a cookie has added the “roll over in the context of the spoken word ROLL” behavior to its appetitive behaviors, which sets up the consummatory behavior of cookie-eating.
- **Animals learn the value of stimuli and behaviors:** Stimuli that identify motivationally significant contexts, and appetitive behaviors that lead to goal-satisfying contexts, both have learnable values associated with them. This is an underlying assumption of many associative models of classical or operant conditioning [Sutton90, Klopf93, Schumujuk93, Montague94, and LeDoux90]. It is also a necessary assumption for integrating learning into an ethologically inspired model, such as that used in Hamsterdam, in which external and internal factors are expressed as real-valued quantities.
- **Animals learn more than associations:** It is not enough simply to learn that some action X is a useful appetitive behavior. It is also useful to learn an expected time course for X, that is, how long to engage in action X before concluding that reinforcement is not forthcoming and that another behavior may be more successful. Gallistel [Gallistel90,95] presents substantial evidence that certain animals in certain circumstances can learn the information relevant to this kind of time course knowledge, including the time of occurrence of events, the time interval between events, the number of events, and their rate. Within an associative model, such as the one we develop here, it is also necessary to augment the system's knowledge with such additional statistics in order to allow learning of a behavior's expected time course.

There are two keys points here that bear emphasis. The first is that ethologists view learning as occurring within a pre-existing structure, and this structure determines what can be learned and often the manner in which it is to be learned. This means that much of the learning which occurs can be viewed as a reorganization of innate components (i.e. the animal learns through experience that a given stimulus to which it is innately sensitive or an innate motor pattern is relevant to a particular motivational system). This is an approach taken by a number of ethologists including Lorenz [Lorenz73], Hogan [Hogan88], Timberlake [Timberlake89], and Davey [Davey89]. For example, Hogan presents an elegant explanation of the developmental process in chicks along just these lines.

Similarly, Lorenz [Lorenz73] viewed animal training as a special case of this approach to learning. This is illustrated in Figure 3 which contrasts a "behaviorist" view of animal training with an Ethological view. Once again it is important to note that evolution constrains the types of associations which can be learned, and ultimately the possibilities for re-organization (as Lorenz puts it, as only he can, "there are no rat prostitutes" [Lorenz73], i.e. it is next to impossible to induce a rat to make sexual displays for a reward).[11]

Figure 3 Ethological View of Operant Conditioning



Ethologists such as Lorenz (Lorenz73) view operant conditioning (animal training) as largely a process of re-organization. In this case, a horse learns that rearing up (a behavior normally associated with defense) when the trainer speaks (a new context) is a useful appetitive strategy for finding food. Conceptually this means that the "ground predator" behavior and its associated motor action is copied into the behavior system associated with feeding. In addition, a new Releasing Mechanism (i.e. detects trainer's command) is added to the copy of the behavior in the feeding system.

The second point is that because learning typically occurs within a structure which often pre-disposes the animal to be sensitive to relevant stimuli, simple associative learning may underlay what appears to be more complicated learning. Socially-facilitated learn-

11. Our learning model does not currently reflect these types of constraints, but this would be an easy extension.

ing is a case in point. Socially-facilitated learning is a form of learning in which an animal learns from the actions of another animal which, in effect, acts as a teacher. This need not imply imitation, since the animal may in fact be learning to pay attention to the same object to which the “teacher” is attending. As Gould [Gould94], Shettleworth [Shettleworth94], Lorenz [Lorenz73] and Plotkin [Plotkin93] point out, this learning often takes the form of associative learning under the direction of a “learning program”. For example, baby birds upon hearing the “mobbing cry” will associate it with any large foreign object in their environment, and will subsequently “mob” that object. Since the object is typically a predator, this “innate learning mechanism” (Lorenz’s term) provides them with a easy way to learn what a predator is. Yet, at its root, it is simple associative learning. Or, for example, rats generally avoid novel foods (for good reason given their historical relationship to humans), but will sample a novel food if they have previously smelled the food on the breath of a living rat [Shettleworth94]. Similarly, baby vervet monkeys have an innate pre-disposition to use a vocalization reserved for signalling the presence of aerial predators whenever they see a larger flying bird. However, by watching the reaction of older monkeys to their call, they learn to discriminate between true aerial predators (eagles) and other large but harmless birds (vultures) [Cheney90]. Baby beavers learn to identify predators and signal appropriately in a very similar manner [Hodgdon78]. Finally, one can imagine a simple form of imitation which may be innately triggered by the presence of a predator which basically says: “In danger, do what Mom does”. Each case is illustrative of perhaps simple associative learning embedded in an innate structure. In other words, through evolution animals develop innate mechanisms which facilitate learning largely by addressing the specific structural and temporal credit assignment problems which are relevant to the particular lesson to be learned.

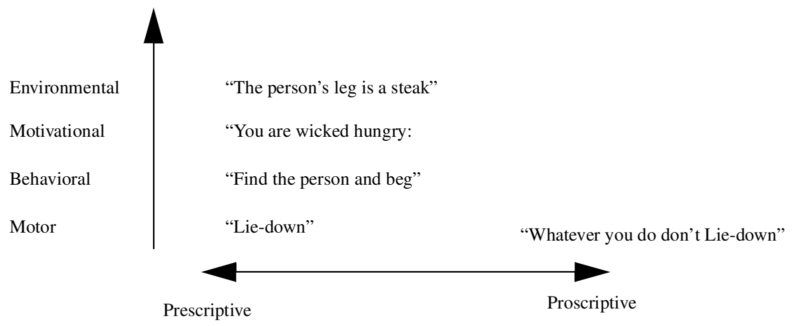
Socially-facilitated learning is a particularly interesting type of learning because it allows animals to learn relatively complex lessons from the actions of other animals by using a simple learning model which is triggered at just the right instant, and which attends to the “right” stimuli. There need not be intent on the part of the teacher, nor feedback or communication (all hard problems for robots). More generally, it represents an excellent example of perhaps the most important lesson for learning from ethology: a simple learning mechanism can suffice if embedded in the right structure.

4.7 Classical Animations Perspective on External Control

The discussion up to this point has assumed that pure autonomy is the goal. While this is appropriate for certain applications, in many applications one may wish to augment the autonomous behavior with external direction so as to better achieve the objectives of the application. One can distinguish at least four levels of abstraction at which one might wish to externally control an autonomous creature (see Figure 4). These are:

- **The motor skill level**. Here the director wants the character to engage in specific low-level actions (e.g. “Go-Forward”, “Look-At”). These actions may be “in-character” (e.g. “Walking”) or “out-of-character” (e.g. “SetPosition”). Typically, no planning or elaborate sequencing is required by the character, nor is any sensory input used.
- **The behavioral level.** Here the director wants the character to engage in a specific, albeit complicated behavior (e.g. “Find that juicy steak in the user's hand”), and

Figure 4 Multiple Levels of Control



trusts that the character will do the right thing which may require some sort of planning or sequencing of actions by the character. Typically, the character must use sensory input to accomplish the task.

- **The motivational level.** Here the director is not necessarily directing the character to engage in a specific behavior, but rather is predisposing it to act in that manner, if other conditions are right (e.g. “You are very hungry”). For example, if no food is present the character may not necessarily search for food, but the moment it catches a whiff of a steak, it heads directly for it.
- **The environmental level.** Here the director manipulates the behavior of the character by manipulating its environment, thus inducing a potential response from the creature (e.g. “There is a juicy steak on the table and your owner is outside”). Imaginary sensory input is a special case of this type of control in which the director manipulates the sensory input of the character so that the character effectively “imagines” the existence of a significant object or event and acts accordingly. For example, if the director wants the character to approach the user, the director could attach an imaginary “dog biscuit” to the user's hand that is only visible to the dog.

Rather than viewing external control as “directorial dictates”, we view it more as a mechanism for expressing “weighted preferences or suggestions” for or against certain behaviors or actions. The higher the weight, the greater the preference’s influence on the creature's subsequent behavior. Seen in this light, a preference is just one more factor in the mix of internal motivations and external stimulus which ultimately determine in which behaviors the creature will engage.

4.8 Summary

In this section we have discussed what we saw to be a few of the key lessons to be learned from ethology and classical animation. We saw that ethology is a valuable source for ideas because ethologists address relevant problems, have a bias toward simple non-cognitive explanations, and tend to analyze behavior at the same level which we wish to synthesize it. Classical animation, while focusing on observable behavior as well, serves as a source for ideas on how we intuit motivational state and intentionality from motion. We saw that ethologists believe that individual behaviors are responsible for assessing their own relevance, and do this by weighing relevant internal and external factors. We saw that coherence was seen as arising from the mechanism by which behaviors compete (i.e. mutual inhibition), and out of the larger organization (i.e. cross exclusion groups). But we also saw that the need for coherence needs to be balanced by the need for "losing" behaviors to express their preferences for motor actions, and this suggested that the underlying motor system needed to support coherent, concurrent motion. This was important not only from a computational perspective but also from the perspective of classical animation, where motivational state and intentionality is conveyed through motion. There were two key messages from ethology with respect to learning. The first message was that in animals, learning takes place within a structure which makes it easy to learn what they ought to learn. The second message was that an important type of learning consists of discovering that an existing behavior when performed in a novel context leads to the satisfaction of a previously unassociated motivational variable. Finally, we suggested that any architecture for autonomous animated creatures needed to incorporate a way of integrating external control, and we saw that there was "no right level" of external control, and it needed to be integrated at several different levels.

In the next chapter we present our computational model which incorporates these ideas.

5.0 Computational Model

In this section we describe a computational model for interactive animated creatures which builds on the ideas from ethology and classical animation that were presented in the previous section. This model has been implemented as part of our ethologically inspired tool kit for building autonomous animated creatures (i.e. Hamsterdam) and we have used it to construct a number of autonomous animated creatures, including Silas T. Dog, the virtual dog featured in the ALIVE project. Through the process of constructing these creatures and observing their resulting behavior in a variety of environments, we have gained insight into how each of the problems described in section 3.0 may be addressed and solved via a few important ideas from ethology and classical animation.

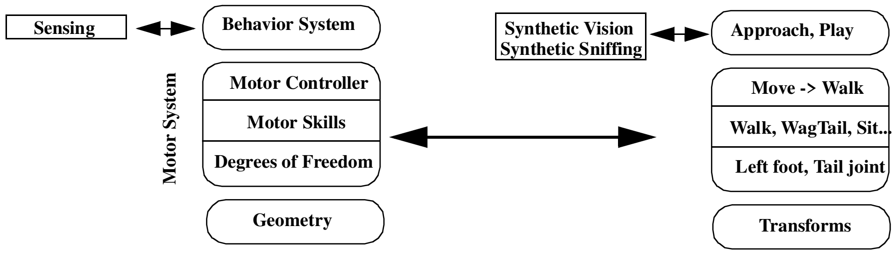
This section is organized around the five big problems of section 3.0 since specific aspects of the model address each of the problems. After a short overview of the architecture, we begin by describing the structure of individual behaviors, the constructs they rely on to determine their value, and ultimately how they determine their own relevance given external and internal factors. The reader will see that our model follows very closely from ethology. Next is a discussion of how Behaviors are organized into groups of competing Behaviors called Behavior Groups, and the details of how they compete using mutual inhibition and boredom.We will also discuss how this model explicitly addresses the problem of persistence. We will then turn to a discussion of the design of the underlying motor-system and show how it allows multiple behaviors to express their preferences for motor actions while still insuring coherent, concurrent motion. This is an essential part of enabling a creature to convey its motivational state and intentionality through its movement. We then summarize the action-selection algorithm. At this point we shift focus a bit and describe how learning is integrated into this approach, once again borrowing heavily from ethology. We then describe how external control can be integrated as well. Lastly, we describe our approach to sensing and in particular our use of synthetic vision for low level navigation and obstacle avoidance.

Before turning to the specifics of the computational model, it is important to understand the basic computational structure of our interactive creatures[12]. A creature in our system (see Figure 5) consists of the three basic parts (Geometry, Motor Skills and Behavior System) with two layers of abstraction between these parts (Controller, and Degrees of Freedom).The Geometry provides the shapes and transforms that are manipulated over time to produce the animated motion. The Motor Skills (e.g. “walking,” “wagging tail”) provide atomic motion elements that manipulate the geometry in order to produce coordinated motion. Above these Motor Skills sits the Behavior System. The purpose of the Behavior System is to send the “right” set of control signals to the motor system at every time-step. The “right set” is determined by weighing the current competing goals of the creature, assessing the state of the environment, and choosing the behavior (and its associated set of actions) that best satisfies some of the creature's goals at this instant in time. More generally, the Behavior System coordinates the creature's use of its available high-level behaviors in a potentially unpredictable environment.The Behavior System triggers the correct Motor Skills to achieve the current task or goal. Between these three parts are two layers of insulation, the controller and the degrees of freedom

12. Perlin’s [Perlin96] architecture for creatures utilizes a similar decomposition of function.

(DOFs) which make this architecture generalizable and extensible. The DOFs, Controller, and Motor Skills together constitute what we call the Motor System.

Figure 5 Architecture



Our architecture is composed of 5 basic layers as shown above. The geometry corresponds to the transforms, shapes and materials which make up the creature. The Degrees of freedom represent "lockable knobs" which may be used to modify the geometry. DOFs may correspond to a joint, or an articulated body manipulated via inverse kinematics. Motor Skills utilize the DOFs to produce coordinated "atomic" motion over time. The Motor Controller maps abstract motor commands onto calls to the particular Motor Skills which implement that functionality for the creature. The Behaviors which make up the Behavior System compete for control of the creature, i.e. for the prioritized right to issue motor commands to the controller.

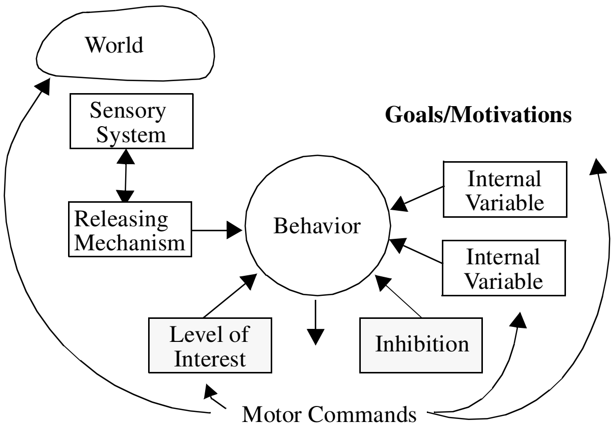
5.1 Behaviors, Releasing Mechanisms, and Internal Variables

Our computational model for the Behavior System follows directly from the ethological perspective of a distributed network of self-interested, goal-directed entities called Behaviors[13]. The granularity of a Behavior's goal may range from very general (e.g. "reduce hunger") to very specific (e.g. "chew food"). Each Behavior is responsible for assessing its own current relevance or value, given the present state of internal and external factors. The current value of a Behavior may be high because it satisfies an important need of the creature or because its goal is easily achievable given the current state of the environment. Our conceptual model of a Behavior is summarized in Figure 6

A Behavior competes on every time step (1/20 of a second) for control of the creature on the basis of its value. Winning the competition means that the Behavior has prioritized access to the creature's resources. This means for example that it has first crack at issuing motor commands. Losing behaviors may still issue motor commands, but they are only executed if they do not conflict with those issued by the winning Behavior. If a Behavior is very general (e.g. "reduce hunger"), winning means that the Behavior's children (i.e. other, more specific Behaviors which help achieve the general Behavior's goals) are given a chance to compete for control and so on. Behaviors influence the system in several ways: by issuing motor commands which change the creature's relationship to its environment, by modifying the value of Internal Variables, by inhibiting other Behaviors, by issuing suggestions which influence the motor commands issued by other Behaviors, and by deciding which other Behaviors have a chance to influence the system.

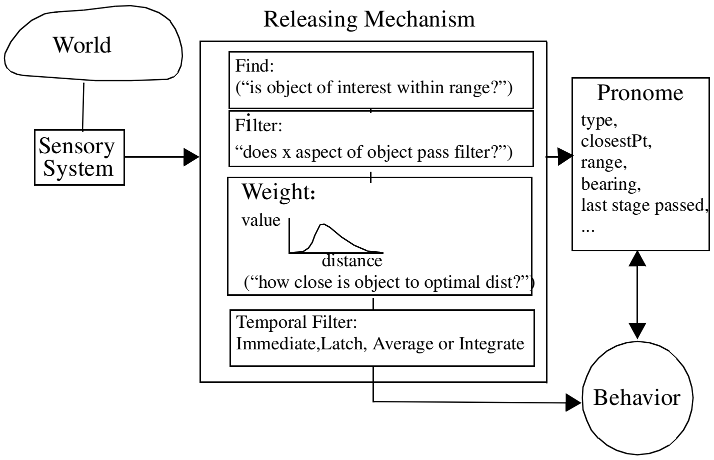
13. We will use the convention of capitalizing the first letter of constructs which correspond to actual classes in Hamsterdam. For example, Behaviors, Releasing Mechanisms, Internal Variables, Motor Skills, Degrees of Freedom and Behavior Groups are implemented as classes in Hamsterdam.

Figure 6 Each Behavior is responsible for assessing its own relevance



The purpose of a Behavior is to evaluate the appropriateness of the behavior, given external stimulus and internal motivations, and if appropriate issue motor commands. Releasing Mechanisms act as filters which identify significant objects or events from sensory input, and output a value which represents the strength of the sensory input. Motivations or goals are represented via Internal Variables which output values which represent their strength. A Behavior combines the values of the Releasing Mechanisms and Internal Variables on which it depends and that represents the value of the Behavior before Level of Interest and Inhibition from other Behaviors. Level of Interest is used to model boredom or behavior-specific fatigue. Behaviors must compete with other behaviors for control of the creature, and do so using Inhibition.There are a variety of explicit and implicit feedback mechanisms.

Figure 7 Releasing Mechanisms identify significant objects or events from sensory input



Releasing Mechanisms identify significant objects or events from sensory input and output a value which represents the strength of the stimulus. By varying the allowed maximum for a given Releasing Mechanism, a Behavior can be made more or less sensitive to the presence of the relevant stimuli. A Releasing Mechanism has 4 phases (Find, Filter, Weight and Temporal Filtering), as indicated above. Temporal Filtering is provided to deal with potentially noisy data.

Behaviors rely on two abstractions to help determine their value. The first, Releasing Mechanisms, are used to detect relevant stimuli in the environment and transduce a

Figure 8 Example Definition of a Releasing Mechanism

```
CreatureFieldRM {
    type        PERSONTAG
    logicalOp   IF_ANY
    filterFields [extendedLeft, extendedRight]
    activeRange [0 5 20]
    maxValue 20.0
}
```

Above we show an example of the code used to define a Releasing Mechanism. In this case the RM is active when it detects a person within 20 units and the person's hand is extended. Its maximum value is 20, and is reached when the person is 5 units away. Note, "extendedLeft" and "extendedRight" are Boolean fields which the Person object makes available for interrogation by other creatures.

value. The second, Internal Variables (sometimes referred to as Motivational Variables) are used to model the effects of internal state such as hunger, or thirst. These abstractions are described below.

5.1.1 Releasing Mechanisms and Pronomes

Behaviors rely on Releasing Mechanisms (see Figure 7) to filter the creature's sensory input so as to identify objects and events whose presence helps determine the current appropriateness of the Behavior. Following from the Ethological argument for using a common currency, Releasing Mechanisms output a continuous value that typically depends on:

- the presence of the stimulus object or event (e.g. a person is nearby).
- more specific features associated with the stimulus (e.g. the person's hand is down and extended).
- the distance to the stimulus relative to some ideal distance (e.g. the hand is one foot away, and I like hands that are two feet away).

During action-selection a Releasing Mechanism updates its value based on the presence, or lack thereof, of relevant stimuli. This process involves four phases, a find phase which attempts to detect the RM's object of interest, a filter phase which filters the attributes of the object of interest for more specific criteria (e.g. that the hand is low and extended), and a weighting phase which is responsible for weighting the value of the RM based on some criteria such as distance. Typically, the value depends on the distance of the object of interest from the creature: beyond a certain range the value is 0.0, and the value increases to an allowed maximum according to some function as the distance decreases to some optimal distance. It is worth noting that such a weighting scheme encourages persistence, since as the creature gets closer to the object, the greater the motivation to proceed. That said, in practice we have found it useful to have a variety of weighting functions (e.g. linear, exponential, flat, etc.). Finally, Releasing Mechanisms may also do some type of temporal filtering such as averaging or summing over a period or implementing a kind of latch in which the RM is active if the relevant stimuli was detected at any time within a given period.

While Releasing Mechanisms may vary in their specifics, they have a very stereotypical structure and are easily parameterized to deal with the specifics. Thus, in Hamsterdam it is rarely necessary to define a new class of Releasing Mechanism. See Figure 8 for an example of how one defines a Releasing Mechanism in Hamsterdam.

The traditional view of a Releasing Mechanism was as a filter which when activated by appropriate stimuli released goal-driven behavior [Tinbergen53]. In this manner the effect of Releasing Mechanisms was to act as a valve which allowed the "action-specific-energy" to flow into the Behavior. Our view is somewhat different in that the Releasing Mechanism not only identifies relevant environmental situations for the behavior, but also contributes to the behavior's value in the same manner that internal factors do. This view is consistent with the existence of motivational isoclines such as that shown in Figure 1

The maximum value of the Releasing Mechanism is intended to be proportional to the time-discounted usefulness of its object of interest (which its associated Behavior will exploit) to the creature. For example, for a food-sensitive Releasing Mechanism, its maximum value should be equivalent to the time-discounted value of the food source to the creature. We use the time-discounted value to reflect the fact that it will take a finite period of time to consume the food. Thus, with everything else being equal, a food source that provides more energy per time-tick will be favored over one which provides less energy per tick, even if the total amount of energy from the two food sources is the same. This perspective is central to our approach to learning.

Each behavior ultimately relies on one or more Releasing Mechanisms to help it assess its relevance vis-a-vis the creature's environment. Typically, lower level behaviors within a sub-system have more specific releasing mechanisms than those at upper levels of the sub-system. So for example, in a feeding system, a releasing mechanism at high level might fire if food is present, whereas at a lower level a releasing mechanism might fire if the food is moving (and thus activate a stalking behavior), whereas another might fire if the food is not moving (and thus activate an investigatory behavior).

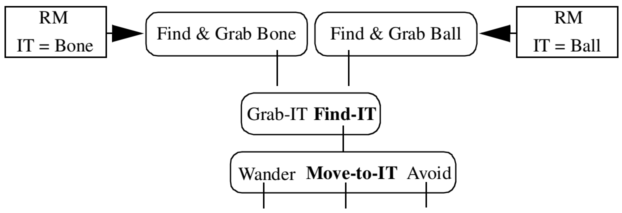
The set of active Releasing Mechanisms within the Behavior System on any given time step is a complex function of the motivational state of the creature, the organization of the various behaviors, and the state of the world. Releasing Mechanisms at the very top level of the behavior system are active (i.e. called on to evaluate the sensory input) on every time step. However, motivational factors play a role in determining to what additional stimuli the creature will attend because of their role in initially determining which sub-systems (e.g. self-defense vs. feeding) will be active. Thus, for example, if the competition at the top level has decided that self-defense is more important than feeding (perhaps because the RMs associated with self-defense in the topmost level suggest a predator is lurking), the host of Releasing Mechanisms associated with the actual behaviors which are responsible for finding food and eating will not be active on that time step. Similarly, attributes of the world as sensed by Releasing Mechanisms at one level may determine what additional features are attended to at a lower level. For example, Releasing Mechanisms at one level may determine that there is an ground predator, and this in turn may mean that the Releasing Mechanisms associated with the behaviors which respond to aerial predators become active, and those associated with ground based predators do not. Thus, attention is a function of which sub-system is active and what behaviors within it are active.

This may seem a bit backwards but the intuition is that we pay attention to what we care about at the moment. For example, if we are hungry we will attend to features of the environment that we associate with food and may "miss" features associated with other motivational systems (i.e. the Releasing Mechanisms associated with those features never get a chance to fire). Similarly, if we are fearful we may hear “noises” that we would pay no attention to otherwise. Thus, our motivational state has a strong influence over what we pay attention to, but it is not an exclusive influence. This is the point of having Releasing Mechanisms at the top level which are always active. An ethologist would say that we attend to just enough features associated with other motivational systems that we can behave opportunistically and, in effect, switch our attention.

5.1.1.1 Pronomes

In addition to transducing a value from sensory input, a Releasing Mechanism also fills in a data structure called a Pronome [Minsky86]. A Pronome can be thought of as a context associated with an “object of interest.” An upper-level Behavior can set the Pronome (e.g. “the red apple”) that lower-level Behaviors then operate on. Specifically, the Releasing Mechanism associated with the winning Behavior at any level makes its associated Pronome be the current object of interest. The Releasing Mechanisms beneath it in the hierarchy are free to use it as their object of interest as well, or ignore it. The practical advantage of Pronomes is that they reduce the size of the Behavior System by allowing portions of it to be shared. These shared portions act as behavioral subroutines which act on the current object of interest as specified by the Pronome. Thus, the same low-level “bite” Behavior may be used whether the Pronome object is a dog-bone or a hamster or the user's kneecap (although with slightly different dentition details). This is

Figure 9 Pronomes allow sub-systems to be shared



Pronomes play an essential role in allowing portions of the behavior system to be shared. In the example above, the behaviors in the “Grab-IT, Find-IT” behavior group and their children are constructed so as to operate on whatever object happens to be the current object of interest (e.g. the object specified by the pronome). This behavior group in turn is shared by two behaviors whose respective Releasing Mechanisms set the pronome to their specific object of interest (i.e. they bind IT to their object of interest). In effect, pronomes allow portions of the behavior system to be used as behavioral subroutines which act on the current object of interest. The behavior group is shared in the sense that it will be active if either of the upper level behaviors are active.

described in more detail in Figure 9

Pronomes as shareable structures run counter to the traditional ethological (and autonomous agent) approach of self-contained, independent behaviors but when thought of as representing the current focus of attention they make a great deal of sense. They also make sense from the standpoint of an efficient implementation. Without Pronomes, one would need a different set of behaviors for each specific kind of object on which they acted (e.g. “approach-ball and approach-bone” and “approach-sock” would all need to be implemented as separate behaviors even though they are virtually similar except for the object which is to be approached).

5.1.2 Internal Variables

Internal Variables are used to model internal state such as level of hunger, thirst, fear, or aggression. Like Releasing Mechanisms, Internal Variables are entities which output a continuous value on every time step. However, in the case of Internal Variables, the change is the result of internal factors such as:

- Autonomous growth and damping over time (e.g. hunger increases over time subject to specific growth and damping rates)
- A side-effect of a behavior being active (e.g. when the "chewing" behavior is active, it reduces the level of hunger by an amount proportional to the value of the behavior). In Ethological terms, "chewing" is an example of a "consummatory" behavior, which when active has the effect of reducing the further motivation to perform that behavior.
- A side effect of a Releasing Mechanism being active (e.g. sensing a snake immediately sets the level of "fear" to be high, which then dissipates over time).

Since Internal Variables are intended to be general abstractions for the end-results of potentially complicated internal systems their value may be the result of an arbitrarily complex calculation. However, in the default case its value may be calculated as shown in Figure 10 In our default case, the value of an Internal Variable goes up over time

Figure 10 The Internal Variable Update Equation

$$IV_{it} = (IV_{i(t-1)} \cdot damping_i) + growth_i - \sum_k effect_{kit}$$

where:

IV_{it} = Value of Internal Variable i at time t

$damping_i$ = Damping rate associated with Internal Variable i

$growth_i$ = Growth rate associated with Internal Varible i

$effect_{kit} = modifyGain_{ki} \cdot Value_{k(t-1)}$

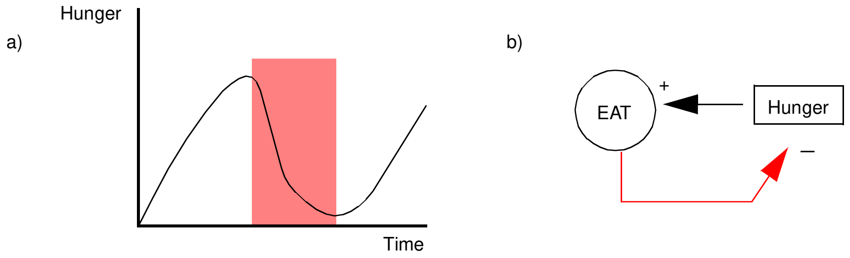
$effect_{kit}$ = Adjustment in value due to behavior k at time t

$modifyGain_{ki}$ = Gain used by behavior k to adjust Internal Variable i

$Value_{k(t-1)}$ = Value of behavior k at time t-1

reaching an asymptote at a value equal to growth/damping. When a relevant behavior is active, it will typically reduce the value of the Internal Variable by an amount proportional to the behavior's value. The intuition here is that initially the behavior will have a large effect on the level of the variable, but as time progresses the behavior has less of an effect. The reader will note that our default case most closely models a homeostatic variable. That is, a variable which rises over time until a relevant consummatory behavior becomes active which has the side-effect of reducing the homeostatic variable back toward some set point. This makes them convenient for modeling, at least at a first approximation, the effect of internal factors such as hunger or thirst. However, this does not mean that they can't be used model the effects of other types of internal factors. For

Figure 11 Motivational Variables



The value of Motivational Variables typically change over time. For example, in (a) we show how the value of a motivational variable such as hunger might change over time. Hunger has an endogenous level of growth which causes it to grow over time anytime it isn't eating. At some point, the combination of hunger and appropriate stimulus (i.e. food) causes the eating behavior to become active (shown as the shaded area above), and this in turn causes the level of hunger to fall, and eventually the creature stops eating, at which point hunger begins to rise again, and so on. As shown in (b) a behavior's value may depend on the value of one or more motivational variables. In addition, there maybe a feedback relationship such that an active behavior will reduce the level of a motivational variable upon which it depends by an amount proportional to the current value of the behavior.

example, as pointed out above, one could have a "Fear" variable whose value was explicitly set by a Releasing Mechanism, and then allowed to decay over time[14]. Or for example, a variable could represent "Frustration" and grow as a function of the length of time a behavior was active. Indeed, if there was a cognitive component to the Behavior System it might set a variety of variables in response to its assessment of the current situation. Thus, our model of Internal Variables is purposely left very general so as to accommodate a variety of alternatives. From the standpoint of the other components in the Behavior System the only important thing is that an Internal Variable returns a value when queried.

Internal Variables typically have a more global effect on the system than do Releasing Mechanisms. Whereas Releasing Mechanisms signal the appropriateness of a specific behavior (or small group of behaviors), the effect of an Internal Variable is usually to predispose or bias the creature toward a certain course of action, without specifying the exact behavior which is to be performed on a given instant. For example, a high level of hunger will predispose the creature to engage in behaviors which will reduce its level of hunger, but it is usually a Releasing Mechanism which determines which of several alternative strategies is best at any given instant. As we will see in 5.2, Behaviors tend to be organized in a loose hierarchy, and it is in the upper levels of the hierarchy that Internal Variables typically have their greatest effect (e.g. does the creature engage in reducing its level of hunger or in mating). Barring unexpected opportunities as signalled by Releasing Mechanisms associated with Behaviors in the upper levels of the hierarchy, choices at this level tend to be largely influenced by Internal Variables.

14. Indeed, one could argue that one of the differences between modeling emotions and motivations is that emotions, particularly in animals, may have a strong reactive component (i.e. an event in the environment may cause a rapid change in a given emotion), whereas motivations tend to change slowly over time in response to internal changes.

If Internal Variables as proxies for motivations influence which behaviors are performed, Internal Variables as proxies for "emotions" can influence how they are performed. Often, the emotional state of a creature may not affect what they do so much as how they do it. A fearful cat may still approach the door to come in, but will do it in a different manner than a bold cat. As we will see in a later section, the system is designed so that even behaviors that lose the competition for control may still influence what happens by issuing suggestions as to what motor commands should be performed or how they should be performed. In practice, we use Internal Variables as proxies for "emotion" to influence what suggestions these "losing" behaviors make. That is, the behaviors are written so that the suggestions they issue are based on the level of the relevant Internal Variables (e.g. the "move-ear" behavior in Silas sets the ear position in direct proportion to the level of the "Happiness" Internal Variable).

As we will see later, motivations (i.e. Internal Variables) play an important role in learning. Indeed, much of the learning that occurs in animals centers around the discovery either of novel strategies or situations which reliably lead to the satisfaction of one or more motivational variables. We adopt just such a perspective in our model: Internal Variables, our system's representation of motivations, are viewed as entities that actively drive the attempt to discover new strategies (i.e. appetitive behaviors) that insure their satisfaction. Specifically, we use the change in the value of a motivational variable due to the activity of a consummatory behavior as a feedback (or reinforcement) signal for the learning process.

Figure 12 shows an example definition of a Behavior and that of the Releasing Mechanism and Internal Variable upon which it depends. As the example suggests a Behavior typically depends on one or more Releasing Mechanisms and Internal Variables to determine its value. During action-selection a Behavior's Releasing Mechanisms and Internal Variables are queried for their current value, and then the value of the Behavior

Figure 12 Example Definition of a Behavior

```
DEF DESIRE_TO_PLAY InternalVariable {
        pname "DESIRE TO PLAY"
        initialValue 1.0
        growth .1 damping .01
}
DEF SIT Behavior {
        leafCBName "sitCB"
        DependsOnVariables {
                USE DESIRE_TO_PLAY
                DEF HAND_DOWN CreatureFieldRM {
                        type            PERSONTAG
                        logicalOp IF_ANY
                        filterFields [extendedLeft, extendedRight]
                        activeRange [0 5 20]
                        maxValue 20.0
                }
        }
        ModifyVariable { who DESIRE_TO_PLAY gain -.01}
}
```

Above we show a fragment of a behavior system for a creature such as a dog. It defines an Internal Variable named PLAY which has an initial value of 1.0 and autonomous growth nd damping rates of.1 and.01 respectively. The SIT behavior rely on the state of the DESIRE_TO_PLAY internal variable and the HAND_DOWN releasing mechanism. When SIT is active, it reduces the level of the DESIRE_TO_PLAY internal variable by -0.01 times its value.

is calculated using the behavior update equation summarized in Figure 13. McFarland

Figure 13 The Behavior Update Equation

$$V_{it} = \left\lfloor \left[\text{LevelOfInterest}_{it}(\;) \cdot \text{Combine}\left(\left(\sum_k RM_{kt}\right), \left(\sum_n IV_{nt}\right)\right)\right] - \sum_m N_{mi} \cdot V_{mt} \right\rfloor$$

where:

V_{it} = Value of Behavior i at time t

RM_{kt} = Value of Releasing Mechanism k at time t

IV_{nt} = Value of Internal Variable n at time t

N_{mi} = Inhibitory Gain that Behavior m applies against behavior i

Note, index k ranges over set of relevant Releasing Mechanisms for Behavior i, index n ranges over the set of relevant Internal Variables, and m over the set of competing Behaviors in the Behavior Group which holds Behavior i.

argues that the manner in which a given behavior combines relevant internal and external factors to arrive at a value is determined by natural selection and may vary depending on the specific behavior system and the animal's ecological niche. As McFarland puts it: “the modern approach is to regard this question as an entirely empirical matter, not subject to any particular doctrine” [McFarland93]. Indeed, in our model and work we support at least 3 different rules for combining internal and external factors (addition, multiplication, and a mixed rule).[15]

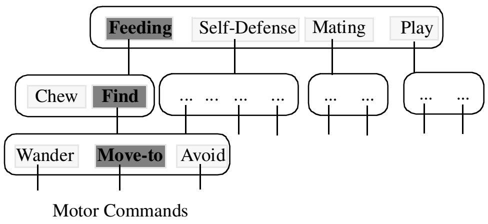
However, as can be seen from the update equation, a Behavior’s ultimate value also depends on the level of inhibition it receives from the Behaviors with which it competes. For an understanding of the exact mechanism by which behaviors compete we must first describe how the system deals with the problem of coherence.

5.2 Behavior Groups and How Behaviors Compete

Following from both the Ethological perspective on how Behaviors are organized and from Minsky’s idea of cross-exclusion groups, Behaviors are organized into groups of mutually inhibiting Behaviors called Behavior Groups. These Behavior Groups are in turn organized in a loose heterarchical fashion as shown in Figure 14. Behavior Groups at the upper levels of the heterarchy typically contain general types of behaviors (e.g. “engage-in-feeding”) which are largely driven by motivational considerations, whereas lower levels contain more specific behaviors (e.g. “pounce” or “chew”) which are driven more by immediate sensory input. The arbitration mechanism built into the algorithm insures that only one Behavior in a given Behavior Group will have a non-zero value after inhibition. This Behavior is then active, and may either issue primary motor com-

15. This flexibility is important, especially when combined with the flexible nature of how Internal Variables may be used. For example, some models of emotions view them as amplifiers which make the creature more or less likely to respond to particular stimuli. This can be easily modeled by assuming a multiplicative relationship between the given emotional variable and Releasing Mechanism.

Figure 14 Behaviors are organized in groups of competing behaviors



mands, or activate the Behavior Group which contains its children Behaviors (e.g. “search-for-food”, “sniff”, “chew” might be the children of “engage-in-feeding”). In Figure 14, for example, the dark gray behaviors represent the path of active Behaviors on a given tick. Behaviors which lose to the primary Behavior in a given Behavior Group may nonetheless influence the resulting actions of the creature by issuing either secondary or meta-commands (these behaviors are shown in light gray in the figure).

Conceptually, the Behavior Group at the top level of the heterarchy typically represents competing goals of the creature (e.g. “do I eat?”, or “do I mate?”), whereas Behavior Groups at lower levels can be thought of as containing alternative strategies for addressing the goal of their parent. For example, a Behavior Group associated with “feeding” might have one “eat” behavior, and a variety of alternative ways (e.g. “find food bowl and rattle it”, or “find garbage can and tip it over”) for getting the creature into a position in which “eat” can become active. Ethologists call these alternative strategies “appetitive behaviors”, and a behavior like “eat” an consummatory behavior. In our system, however, we make no explicit distinction between appetitive and consummatory behaviors other than to insure that the consummatory behavior will “win” in the case of when it and one or more of its associated appetitive behaviors could potentially be active. This can be done in several ways, but the most convenient way is to restrict the range of the consummatory behavior’s RM so that the stimulus has to be within x units of the sensor for it to fire. Similarly, the distance weighting for RMs associated with the appetitive behaviors reaches a maximum outside that limit and decreases to zero as the distance approaches the limit.

As we will see in our section on learning, much of the emphasis is focused on discovering new “appetitive behaviors” which arise from re-using existing behaviors in a next context.

Since each Behavior in the Behavior System is responsible for evaluating its own relevance, information flows into the system at all levels via the Releasing Mechanisms and Internal Variables associated with the individual behaviors. During the action-selection process the winning Behavior in each Behavior Group will post its pronome to a global queue. This is then accessible to Behaviors further below in the system. As mentioned earlier this makes it easy to write generic “move-to and do” sub-systems.

Figure 15 Example Behavior Graph

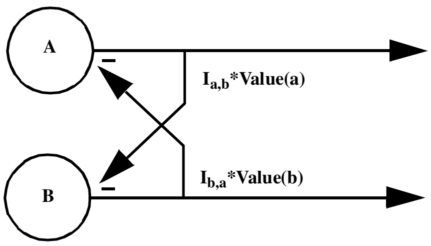
```
DEF WANT_TO_PLAY InternalVariable {
        pname "WANTTOPLAY"
        value 1.0
        growth .1 damping .01
}
BehaviorGroup {
        DEF FINDPERSON_AND_SIT Behavior {
                DependsOnVariables { USE PLAY}
                BehaviorGroup{
                        DEF SIT Behavior {
                                leafCBName "sitCB"
                                DependsOnVariables {
                                        CreatureFieldRM {
                                                type            PERSONTAG
                                                logicalOp IF_ANY
                                                filterFields [extendedLeft, extendedRight]
                                                activeRange [0 5 20]
                                                maxValue 20.0
                                        }
                                }
                                ModifyVariable { who WAN_TO_PLAY      gain -.01}
                        }
                        DEF APPROACH Behavior {
                                leafCBName "approachCB"
                                DependsOnVariables {
                                        CreatureRM {
                                                type PERSONTAG
                                                activeRange [5 10 100]
                                                maxValue 20.0
                                        }
                                }
                        }
                        DEF HOWL Behavior {
                                leafCBName "howl"
                                DependsOnVariable { InternalVariable {value .01}}
                        }
                }
        }
        DEF SOMETHINGELSE Behavior {
        ...
        }
}
```

Above we show a fragment of a behavior system for a creature such as a dog. There is a Behavior Group which has 2 Behaviors (FINDPERSON_AND_SIT, and SOMETHINGELSE). FINDPERSON_AND_SIT's value depends on the value of the WANT_TO_PLAY InternalVariable. It has 3 children behaviors in its Behavior Group (SIT,APPROACH, and HOWL). APPROACH depends on a Releasing Mechanism which fires on the presence of a person between 5 and 100 units, and which reaches its maximum value of 20 when the person is 10 units away. SIT's RM is active when the person is within 20 units and a hand is extended. Its maximum value is 20, and is reached when the person is 5 units away. When SIT is active, it reduces the level of the WANT_TO_PLAY internal variable by -0.01 times its value. This in effect reduces the desire of the dog to play. HOWL is active when neither APPROACH or SIT is active but the creature's level of WANT_TO_PLAY is higher than whatever variables and Releasing Mechanisms SOMETHINGELSE depends on.

We show an example definition of a Behavior Group in Figure 15

Within a Behavior Group, Behaviors compete using the Minsky/Ludlow model of mutual inhibition as shown in Figure 16 That is, every Behavior B_i within a Behavior Group inhibits every other Behavior B_j by an amount equal to an inhibitory gain I_{ij} times its value V_i. While by default the inhibitory gain is 2.0, a Behavior is free to use a unique gain I_{ij} for each Behavior it inhibits so long as the chosen gain is greater than 1.0. This model, coupled with clamping the value of Behaviors to be zero or greater, issues that via the "avalanche effect", one Behavior in a Behavior Group will have a non-zero value

Figure 16 Mutual Inhibition is used to provide robust arbitration and persistence



Value = MAX((LevelOfInterest() * combine(RM,IV) - Σ Inhibition), 0)

after arbitration and the others will be zero. As Minsky puts it: “as each competitor grows weaker, its ability to inhibit its challengers also weakens. The result is that even if the initial difference between competitors is small, the most active agent will quickly “lock-out” all the others” [Minsky 86]. Equally important is the fact that once active, a behavior X will stay active until its competitor Y’s value (before inhibition) is greater than $I_{x,y}$*value(X). This, in effect, provides active behaviors with a form of inertia.

To give the reader some insight into the dynamics of switching between two behaviors using the arbitration mechanism described above we give a worked example in TABLE 1. In this example, there are two behaviors (V1 and V2), both of which utilize an inhibitory gain of 2.0. Initially at tick t-1, V2 is active with a value of 10 and V1’s value is 0. However, at tick t V1’s value rises to 21 (perhaps due to the appearance of relevant stimuli) and it is this change in value which prompts the switch in behaviors. Note, for the switch to have occurred, the value of V1 had to exceed 2.0 times V2’s value (i.e. V2’s value multiplied by inhibitory gain used by V2 against V1). On a given tick, the system may have to iterate several times before the system settles. The inhibition on a given iteration is calculated by multiplying the behavior’s value after inhibition on the previous iteration or tick by the appropriate inhibitory gain. The system iterates until only one behavior has a non-zero value. In this example, it takes 4 iterations for the system to settle. However, only one iteration is typically necessary unless the system needs to switch behaviors.

This arbitration mechanism is remarkably robust in practice. However, one does need to deal with certain pathological situations (for example, if V1 and V2 had exactly identical values). Our approach to this and other potential problems is to keep track of the change in value on each iteration. If there is no change and more than one behavior has a non-zero value (i.e. the system isn’t converging), then we pick the behavior with the greatest value (if more than one behavior has the same value we arbitrarily pick one), and for each of the remaining behaviors we set the value they use to calculate inhibition to zero (i.e the value after inhibition on iteration i-1). The effect of this is that only one behavior, the “picked” behavior will spread inhibition on the next iteration. We then restart the iterative process. This biases the system to choose picked behavior.

TABLE 1. Example of Avalanche Effect

	Tick t-1	Tick t Iter. 1	Tick t Iter. 2	Tick t Iter. 3	Tick t Iter. 4	Tick t+1
V1 before Inhibition (iteration i)	0	21	21	21	21	21
V1 after Inhibition (V1 - inhibition from iteration i-1)	0	1	1	5	5	21
Inhibition V1->V2 (used for iteration i+1)	0	2	2	10	10	42
V2 before Inhibition (iteration i)	10	10	10	10	10	10
V2 after Inhibition (V2 - inhibition from iteration i-1)	10	10	8	8	0	0
Inhibition V2->V1 (used for iteration i+1)	20	20	16	16	0	0

Chapter 6 presents results which demonstrate the use of these concepts to control persistence.

As the reader will note from Figure 13 the behavior update equation incorporates a Level Of Interest term (the left-most term) as well. By default, Level Of Interest is set to 1.0, but if desired by the designer, a varying Level Of Interest can be modeled via an Internal Variable whose value is clamped to the range 0 to 1 and which has a multiplicative effect on the value of its associated behavior. Level Of Interest may be based either on time or on some measure of progress as signalled by an associated function which is called on every time step to evaluate progress. Typically, Level Of Interest will decrease over time to 0.0 as the Behavior is active, or if no progress, as measured by the callback, is detected. The callback approach works in applications where there is a clear metric of progress (e.g. in a submarine where there are a number of different strategies for getting back up to the surface, and there is a clear metric of how well you are doing). In other cases, the temporal approach works better (e.g. looking for food). In either case, when the Behavior is no longer active, Level Of Interest will rise back up to 1.0 over time, or if the callback determines that progress may now be made using this Behavior. Additionally, in certain instances a Behavior or Releasing Mechanism will adjust the Level of Interest of another Behavior. See the discussion of motion-energy based navigation in 5.7 for an example of this latter approach.

As noted by Ludlow [Ludlow76], Level Of Interest and this model of mutual inhibition go hand-in-hand. While the use of unique inhibitory gains provides great control over persistence, there needs to be some safety mechanism to prevent one behavior from shutting out the other behaviors while it pathologically pursues an unattainable goal.[16] Level Of Interest plays that role.

The mechanisms described up to this point for managing coherence of action and goal implement a “winner-take-all” system, in which one behavior “wins” and all other behaviors “lose”. As was discussed earlier, it is essential from both the perspective of classical animation as well as ethology, to broaden this view so that (a) even the losers get to express their preferences for motor actions. In addition to providing a mechanism for them to do so, the underlying Motor System must also be designed so that it easily supports coherent, concurrent motion. This is the topic of the next section.

5.3 The Motor System and Coherence of Action

The motor system was designed so that (a) multiple behaviors could express their preferences for motor actions, while (b) nonetheless insuring coherent concurrent motion.[17] This is described in more detail below.

As shown in Figure 5 the Motor System is composed of 3 basic components: the Degrees of Freedom, the Motor Skills and the Motor Controller. The Degrees of Freedom (DOFs) represent lockable "knobs" which are used to manipulate the underlying geometry. Thus, they correspond to entities such as joints (e.g. the tail joint), or collections of joints (e.g. the right front leg). As illustrated in Figure 17, DOFs provide a simple locking mechanism which prevents Motor Skills which work at cross purposes ("Sit" and "Walk") from being active simultaneously.

Motor Skills are entities which use one or more DOFs to produce coordinated motion over time (e.g. "walk", "sit", "wag-tail", "lie-down", "gaze-at"). Each motor skill declares which DOFs it needs in order to perform its task and it can only become active if all of these DOFs are available. Once active, a Motor Skill adjusts its DOFs on each time-step to produce coordinated animation.

Since conflicts among Motor Skills are resolved via the locking mechanism of the DOFs, the Motor Skills may be written fairly independent of each other, with one important caveat. That is, there must be agreement on how to handle transitions between Motor Skills which share DOFs. The approach taken in Silas is that each Motor Skill is responsible for returning its DOFs to some canonical position before it shuts off (e.g. sitting, walking, lying down, and begging all leave Silas in a standing position before they shut off). The advantage here is that the skills need no knowledge of the others. The disadvantage is that transitions aren't as smooth as they could be if a Motor Skill had more knowledge about the other Motor Skills (e.g. if lying down knew how to get from a sitting position to a lying down position, it wouldn't be necessary for sitting to first return the dog to a standing position). In Perlin's system [Perlin96], his Motor Skills equivalents have more interdependencies, but he is able to create nicer transitions as a result.

16. This lesson was brought home to us in ALIVE-1 in which we had a hamster that engaged in a number of behaviors, including foraging for food. This worked out fine when a user was in the space and had food in their hand, because the foraging behavior would cause the Hamster to seek out the user and beg for food. But if there was no user, and the Hamster was hungry then it would only engage in looking for non-existent food and not display its other behaviors, which led to rather uninteresting behavior. Level Of Interest solved this problem nicely, because even if the Hamster was hungry it would temporally lose interest in Foraging and this would give the other behaviors a chance to run.

17. The approach discussed in this section was developed in collaboration with Tinsley Galyean [Galyean95].

Motor Skills are distinguished from Behaviors in two ways. First, Behaviors are goal-directed whereas Motor Skills are not. For example, “Walking” is a Motor Skill whereas “Move toward object of interest” is a Behavior which invokes the “Walking” Motor Skill to accomplish its goal. Second, Motor Skills are similar to the Ethological idea of “Fixed Action Patterns” in that once initiated they do not require sensory input (other than proprioceptive input) and they have a specific time course [Lorenz76][18]. This latter point has great practical significance because it greatly reduces the amount of house-keeping which needs to be done by the behavior system. Specifically, the behavior system never has to explicitly shut off a behavior: it will happen auto-magically (we refer to this as being “spring-loaded to shut off”). For example, a Motor Skill such as “Sitting” is written so that if it was active in the past, but has not been requested on a given tick, it will automatically begin to return the creature to a canonical position (e.g. standing) and will only shut-off once the creature is back in its canonical position. Thus, suppose the creature is sitting, but now wants to start walking. The behavior system will begin telling the motor system to make the “walking” skill active. The “walking” skill will initially be blocked because the required degrees of freedom are still locked by the “sitting” skill. However, because the “sitting” skill is no longer being requested, it begins to move its degrees of freedom back to their canonical position. This will typically take a few ticks, at which point the “sitting” skill releases its Degrees of Freedom and shuts off. Only now can the “walking” skill grab the Degrees of Freedom it needs and become active. This all happens without explicit intervention by the behavior which is issuing the “walking” command. Hence, the advantage of this approach is that the behavior system need only be concerned with what behaviors and motor skills need to be active on a given tick and not with the mechanics of turning off motor skills which are no longer needed. From the standpoint of a given Behavior, it need not concern itself with the effects of other Behaviors (at least at the motor level), and this makes it much easier to create and add new Behaviors

The example in the previous paragraph highlights an important aspect of the interaction between the Behavior and Motor systems. Since action selection happens on every time-step, Behaviors re-issue motor commands on every tick. Typically, they are written to be “stateless”, meaning that the commands they issue are not dependent on how long the behavior has been active. In addition, while a Behavior is told whether it was successful or not in invoking a given motor action (e.g. in the case above, “move-to” is told that “walking” is blocked when it invokes “walking” and “sitting” is still active), Behaviors typically makes no assumptions about the progress of a given motor skill[19]. For example, a “grab-it” behavior will continue to invoke a “grab” motor skill, passing it the updated coordinates of the object to grab, until the Releasing Mechanism associated with another behavior, for example, “move-to-mouth” senses that the object is in fact

18. It should be noted that Fixed Action Patterns as proposed by the classical ethologists are generally seen today as an over-simplification both in terms of the supposed independence from sensory input and in the supposed rigidity of the movements. Nonetheless, they represent a useful design principle.

19. While this is true in our current implementation of Silas, one can easily imagine situations in which a Behavior might care whether it was successful in activating a given Motor Skill. If it wasn’t successful, it might want to choose a different course of action. In general, we believe that the Behavior level is the right place to deal with this problem rather than, say, implementing a queuing mechanism in the Motor System.

grabbed, and that behavior becomes active. Our claim, albeit untested, is that our approach to behavior modeling can be used to control physical systems such as animatronics in which there may be significant noise and unpredictability in the motor system.

However, for our purposes here, the important feature of the Motor Skills and the Degrees of Freedom is that they cooperate to help insure coherent, concurrent motion over time (see Figure 17). This in turn allows multiple behaviors to simultaneously

Figure 17 Motor Skills and Degrees of Freedom insure coherent concurrent motion

Motor Skills	Degrees of Freedom							
Walk	■	■	■	■				
Sit	■	■	■	■	■	■		
Look-At							■	
Wag Tail								■

Degrees of Freedom(DOFs) represent the "knobs" with which Motor Skills modify the underlying geometry over time. To help insure coherent, concurrent motion, only one Motor Skill at a time may have access to a given DOF. A Motor Skill grabs the relevant DOFs when it becomes active, and releases them when it is done. A Motor Skill is blocked from becoming active if another Motor Skill has already grabbed one of the DOFs that it needs. Thus, "Walk" and "Sit" can never be active concurrently because they share DOFs, whereas "Walk" and "Look-At" can be active concurrently.

express their preferences for motor actions while reducing the chances for incoherent motion. But the whole point of the action selection process is, in a sense, to provide prioritized access to the creature's resources, including its motor system. Thus, not only must the Motor System support coherent, concurrent motion, but it must also provide a way to allow behaviors to issue motor commands in such a way that the relative importance of the Behavior (and its associated requests for motor actions) is respected. In our system this is handled by the design of the Motor Controller.

Specifically, the Motor Controller recognizes 3 different imperative forms for issuing any motor command:

- Primary commands (i.e. "Do it")
- Secondary commands (i.e. "Do it if no one objects")
- Meta-commands (i.e. "Do it this way")

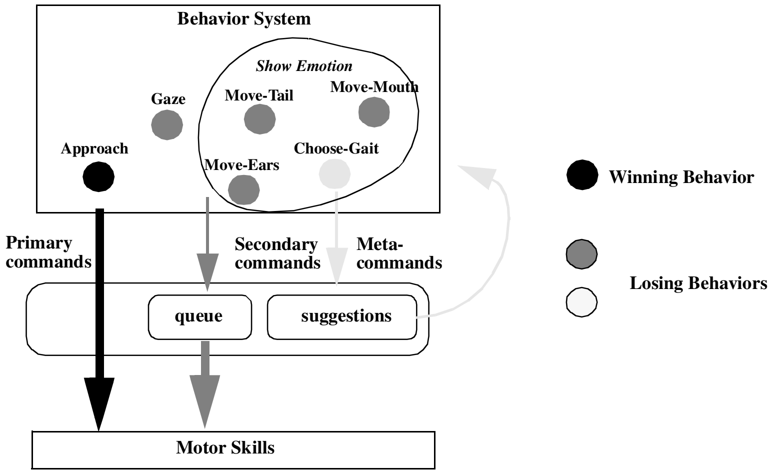
Primary commands are generally reserved for the winning behavior, and are passed on to the Motor Skills right away. By contrast, secondary commands are queued in priority order and executed at the end of the update step assuming the appropriate DOFs are available. The intent is that secondary commands are to be used for suggested but not essential actions. Meta-commands are also queued in priority order but since they are intended as suggestions for other behaviors as to how a given action should be performed (e.g. "Use the following gait, if you are going to move forward"), they themselves are not executed. In fact, meta-commands are really just a mechanism for holding suggested arguments for the motor commands to which they refer. Any motor command can be issued using any one of the 3 imperative forms. The order on the queue in both cases is determined by the value-before-inhibition of the calling Behavior (remember that due to our model of inhibition, the value after inhibition of a losing behavior is always zero). The queues are accessible to all Behaviors, so a winning behavior, for example, may query the queue for a suggestion as to how to move forward (e.g. “move using a walking gait” or “move using a bounding gait”). Similarly, it may remove a queued command if it so desires.

As is described in more detail in Figure 18 winning behaviors typically issue primary commands whereas behaviors which lose typically issue suggestions in the form of secondary and meta-commands as described above. This is implemented in the following manner. Behaviors typically have two callback procedures associated with them and these procedures are actually responsible for issuing the motor commands associated with the Behavior. One, the Behavior’s “action callback” is automatically invoked if the Behavior is the winning Behavior in its Behavior Group. It is in the context of this callback that the Behavior issues the appropriate motor command for when it is active. The other callback, its “suggestion callback” is automatically invoked if the Behavior is among the losers in its group. During action selection the suggestion callbacks associated with the losers are called first, and then the action callback of the winner is invoked. Since the suggestions are posted prior to the winning Behavior taking its action, it can utilize the suggestions as it sees fit. Despite the use of secondary and meta-commands, a winning behavior still has ultimate say over what actions get performed while it is active. It can over-rule a secondary command by removing it from the queue or by executing a Motor Skill which grabs a DOF needed by a given secondary command. In the case of meta-commands, the winning behavior can choose to ignore the meta-command, in which case it has no effect.

We have found the combination of a motor system which supports coherent, concurrent motion, and the mechanism of multiple imperative forms coupled with the convention of allowing losing behaviors to express their preferences for motor actions to be absolutely essential in the design of Silas. For example, in Silas, there are a number of behaviors which are responsible for displaying his emotional state. Thus, there is a Behavior which sets the angle and position of his ears based on the level of the “Happiness” Internal Variable. When Happiness is high the ears are moved forward, and when it is low they are moved back. By construction, these Behaviors are never winning behaviors, and thus they almost always issue secondary commands. This works fine in most cases, but consider the case of the Behavior which chooses a gait based on the level of Happiness. It may not know whether Silas should go forward or not, it only knows how it would “want” Silas to move forward, should he actually move forward. Thus, this Behavior issues a meta command which specifies the desired gait. The great advantage of this approach is that a behavior such as “Approach” need not concern itself with all of the ancillary actions. It simply needs to deal with the problem of navigating to its desired target. If it decides that Silas should move forward, it simply queries the Motor Controller for any suggestions as to the arguments to use when issuing the “move forward” command.

Another important role for the Motor Controller is to provide a layer of insulation between the Behavior System and the underlying Motor Skills so that in certain cases Behaviors can be shared among creatures with different sets of Motor Skills. Behaviors, in fact, do not “talk” directly to Motor Skills. Rather they issue, what may be generic motor commands (e.g. “move-forward”) and the Motor Controller maps them into calls to the appropriate Motor Skills (e.g. “move-forward” maps to “legged-locomotion” in

Figure 18 The Motor Controller supports prioritized expression of preferences



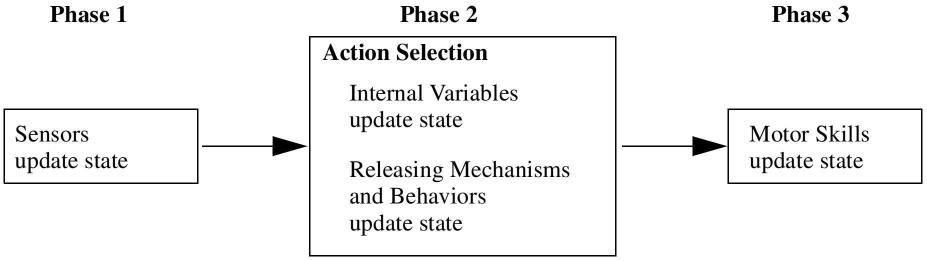
The Motor Controller supports 3 imperative forms for all motor commands. In the example shown above (taken from Silas) the winning behavior is concerned exclusively with approaching a target. There is another behavior whose sole purpose is to gaze at the current object of interest. There is another group of losing behaviors whose purpose it is to indicate Silas's "emotional" state. These behaviors issue a combination of secondary and meta-commands. "Choose-Gait" issues a meta-command because it doesn't "know" if the creature should go forward or not, only the gait to use should some other behavior decide that the creature should move forward.

the case of the dog). Thus, for example, a generic "approach" behavior may be written in terms of "move" and "turn" and used by a wheeled or a legged creature, as long as the Motor Controller of the creatures is able to perform the correct mapping.To be fair, this picture is complicated a bit by the fact that different Motor Skills may require different arguments (e.g. "walking" may take a parameter which is used to specify the actual gait to use). Even here, the design of the motor system is intended to be "user-friendly". As part of the process of constructing a creature, the designer must specify the set of Motor Skills which the creature possesses together with the default parameters to be used when those skills are invoked. In addition, she must specify the mapping from motor commands to Motor Skills. At run-time, as part of the mapping process, the MotorController fills in any arguments which haven't been specified along with the motor command. As long as there is general agreement on the arguments which are shared by all Motor Skills of a particular type (e.g. "speed should be the first argument for any Motor Skill which deals with locomotion"), this default mechanism can help facilitate independence and portability.

5.4 Summary of the Action-Selection Algorithm

Figure 19 summarizes the process by which a creature updates its state on a given time step. There are essentially three phases. In the first phase the creature's sensors update their state. For example, if the sensor utilizes synthetic vision, then it is at this point that the image is acquired. In the second phase, the Behavior System updates its state. This can be thought of as happening in two stages. In the first stage, all of the Internal Variables update their state, reflecting the effect of endogenous change in their value or as a

Figure 19 Update Process



result of a side-effect of a Behavior being active on the previous time step. The actual process of action selection happens in the second stage as follows:

(1) Starting at the top-level Behavior Group, the Behaviors within it compete to become active. This is done as follows:

(1.1) The Releasing Mechanisms of each Behavior update their value based on the current sensory input. This value is then summed with that of the Behavior's Internal Variables and the result is multiplied by its Level Of Interest. This is repeated for all Behaviors in the group.

(1.2) For each Behavior in the group, the inhibition due to other Behaviors in the group is calculated and subtracted from the Behavior's pre-inhibition value and the Behavior's post-inhibition value is clamped to 0 or greater.

(2) If after step (1) more than one Behavior has a non-zero value then step (1) is repeated (using the post-inhibition values as the basis) until this condition is met. The Behavior with a non-zero value is the active Behavior for the group.

(3) All Behaviors in the group which are not active are given a chance to issue secondary or meta-commands. This is done by executing a suggestion callback associated with the Behavior.

(4) If the active Behavior has a Behavior Group as a child (i.e. it is not a Behavior at the leaf of the tree), then that Behavior Group is made the current Behavior Group and the process is repeated starting at step (3). Otherwise, the Behavior is given a chance to issue primary motor commands via the execution of a callback associated with the Behavior.

As a result of action selection, Behaviors will have issued motor commands. These motor commands have the effect of setting parameters in the appropriate Motor Skills. However, the Motor Skills do not do anything (i.e. no geometry is modified) until the third phase of the update process. In this last phase, each Motor Skill is given a chance to update its state. In this phase that the Motor Skills actually modify the underlying geometry in response the issued motor commands. Note, even a Motor Skill which has not been issued any motor commands on a given tick is given a chance to run. This allows a Motor Skill to shut itself off as described earlier.

5.5 How Adaptation is Integrated into the Behavior System

The structures and algorithms described up to this point deal with the issues of: doing the right things, for the right amount of time, while conveying their intentionality and motivational state. But such creatures, while their behavior may be so organized that they are motivated to do what they ought to do at a particular time (to paraphrase McFarland), they will not adapt their behavior based on their experience. We now wish to rectify this problem by showing how adaptation may be integrated into this approach. Specifically, we want a creature to be able to learn:

- that an existing behavior can lead to the fulfillment of some previously-unassociated motivational goals when performed in a novel context;
- new releasing mechanisms which tell the creature something of value different or sooner than existing releasing mechanisms (i.e. they have incremental predictive value);
- appropriate values for existing releasing mechanisms to reflect the perceived value of the signaled object to the creature.

These are three important types of learning for animals. The first of these items is a thinly veiled description of operant conditioning which is thought to underlie much of animal training. That is, the animal learns an association between an behavior and an outcome, and also learns the stimulus configuration upon which this association is contingent. The second type of learning corresponds more closely to classical conditioning in that there is already a built-in association between a stimulus and a response (i.e. a Releasing Mechanism and a Behavior), and the animal learns that one or more novel Releasing Mechanisms have an incremental predictive value vis-a-vis the built-in Releasing Mechanism. The third type of learning corresponds to learning the appropriate value of a stimulus based on experience. For example, learning which of two food sources is a better choice. In the first two cases, the creature may be learning novel Releasing Mechanisms and uses for existing behaviors and altering the structure of its Behavior System as a result. In the last case, the creature is simply refining existing Releasing Mechanisms.

Fundamentally, the animal is trying to learn one of a number of potential associations:

- $((S)\rightarrow(A\rightarrow O))$: Behavior A leads to outcome O, in context S
- $((S_1\rightarrow S_2)$->$(A\rightarrow 0))$: Context S_1 predicts Context S_2 which provides a context in which performance of Behavior A leads to outcome O.
- $((S_1,S_2)\rightarrow(A\rightarrow O))$: Context S_1 and Context S_2 both serve as a context in which the performance of Behavior A leads to outcome O_i, but what is their relative value?

In our model, the stimulus is a feature of the world which changes state and whose change in state can be detected and attended to by the creature. The user moving their hand to a pointing position, or issuing an utterance are two examples of what we mean by stimulus. A change in the state of a feature of the world may be important because it identifies a special context in which if the creature responds in a particular way the world will change in a predictable way (i.e. lead to a new context). In this case, the stimulus is called a "discriminative" stimulus because it allows the creature to discriminate when performance of a behavior will lead to a particular outcome. Releasing Mechanisms are always used to detect stimulus, and when we talk about the "value" of a stimulus, we mean the value output by a Releasing Mechanism which detects and fires on the presence of the stimuli. The creature has a set of built-in Releasing Mechanisms and may create new ones as a result of learning.

The action corresponds to a leaf behavior such as "Sit", "Beg" etc. which issues one or more motor commands when active. It is through these motor actions that the creature affects the world, leading perhaps to a new context.

The outcome corresponds to a reinforcer, that is, something which has value to the creature. A reinforcer may have value because it directly leads to a change in a motivational variable (we will call this an "intrinsic" or "un-conditioned reinforcer"). For example, a cookie may serve as a reinforcer for a hungry animal: when it eats the cookie it becomes less hungry. However, there are other kinds of reinforcers, so-called "conditioned reinforcers" whose value is derived from being associated with the eventual presence of an "intrinsic reinforcer". This can be seen in the following example.

Suppose we want to train the dog to do a sequence of actions: beg when it hears a bell and then sit when it hears a whistle. Typically, an animal trainer would teach the dog the last trick first and then working backwards to the start of the routine[Rogerson92]. To teach the dog to sit when it hears a whistle, the trainer would make sure that the dog was hungry and then do a series of training exercises in which she "whistles" while simultaneously manipulating the dog into a sitting position.[20] Once the dog was sitting, the trainer would reward the dog with a treat. If the dog is suitably motivated, it quickly learn that sitting in response to a whistle reliably leads to a reduction in hunger. The reinforcer is the dog biscuit, the action is sitting, and the stimulus is the whistle. Once it appears the dog has learned this lesson, the trainer would work on the behavior which is to precede this one in the sequence. In this case, the trainer would ring a bell while manipulating the dog into a begging position, and once the dog is begging, respond by "whistling". If the dog has truly learned the first lesson, it will respond to the whistle by sitting and the trainer, in turn, will give the dog a biscuit in response. In this case, "whistling" acts as a conditioned reinforcer whose value is derived from the fact that it signals the context for a behavior which leads to the presence of an intrinsic reinforcer. Thus, one behavior's stimulus may be another's conditioned reinforcer.

What we have described in the previous paragraph is basically operant conditioning in which the creature learns a stimulus-behavior pair which are found to be reliably associated either with a change in an underlying motivational variable or with another stimulus-action pair which eventually leads to a change in the motivational variable. We will refer to this process as learning an "instrumental contingency", i.e. that the performance of a specific behavior in a specific context is instrumental in the production of a reinforcer. The stimulus is in close temporal proximity to the action (typically it precedes it

20. Typically, dog trainers do force the dog into the appropriate position by pushing on the dog's rear-end or by pulling on the chain [Rogerson92, Lorenz94]. Alternatively, the trainer could wait until the dog was starting to perform the desired behavior and then produce the appropriate signal followed by the reward. This latter technique is the one used to train marine mammals, for example.

by some small period of time), and it is this temporal proximity combined with a subsequent reinforcer upon which the creature bases the association. Thus, the trick for the creature is to:

- Notice that a motivational variable has changed.
- Look for behaviors and changes in stimuli which are in close temporal proximity to each other and to the change in the variable.
- Decide whether the association is relevant and reliable (see below) with respect to the subsequent presence of a reinforcer. That is, is it worth remembering?
- Assign a value to the presence of the stimulus which reflects in some manner the value of the context in terms of the change in the motivational variable.

5.5.1 Relevance and Reliability

What do we mean by a relevant and reliable association? First, by association we are referring to the conjunction of a stimulus and a behavior. A relevant association, in this context, is one which occurs some percentage of the time that the reinforcer is present. A highly relevant association is one which is present most of the time that a reinforcer follows. Now just because an association is relevant, it need not be reliable. By reliable, we mean that some percentage of the time that the association is present, the reinforcer follows. The association may not occur very often (e.g. the stimulus may be a rare occurrence), nor may it even be present many times when the reinforcer is, but if when it is present, the reinforcer follows some percentage of the time then it may be worth learning. Note that in this case the association may be a reliable one, without being a highly relevant one.

So what does the creature care about: relevance or reliability? Drescher's approach [Drescher91] was to say, in effect: "lets find relevant associations and then learn how to make them more reliable". The idea here is that a reliable association may be a good strategy for getting the reinforcer, but you may not be able to use it very often. On the other hand, if you have a relevant, but unreliable association, it may pay to discover how to make the association more reliable. For example, is there another factor, which if included, would help discriminate between contexts when the strategy will work or not? In other words, relevance helps guide which associations might be worth paying attention to in order to learn a potentially useful association, but reliability ultimately guides action. This is particularly true in a reactive system in which action is driven in part by the presence of stimuli. If the stimuli is present, the creature needs to respond appropriately, i.e. choose the most reliable strategy. Relevance only tells the creature how often it might get to use that strategy.

A related problem for the creature is to know when an association is causal or coincidental. For example, suppose the creature knows that performing behavior B_1 in context S_1 reliably leads to a reinforcer R_1, or using a shorthand $((S_1 \&\& B_1)\text{-}> R_1)$[21]. Now suppose that S_2 almost always precedes S_1, and that this is true no matter how the creature responds to S_2. Thus, the value of S_2 comes not from its role as specifying a context

21. Note we use 'C' conventions here, "&&" means "AND", and "||" means "OR", (!B) means "NOT B", and "!=" means "NOT EQUAL".

in which performance of a specific behavior will result in S_1, but rather as a predictor of the future presence of S_1 independent of what action the creature takes. In this case, it may pay for the creature to respond to S_2 in the same manner to which it responds to S_1. This may allow it to get more reinforcement or get it sooner because it doesn't have to wait until S_1. That is, the creature should learn that performing behavior B_1 in context S_1 or context S_2 reliably leads to a reinforcer R_1, or using our shorthand $((S_2 \| S_1) \&\& B_1) \rightarrow R_1$).

The problem which arises in animals and animats alike is that since the creature is always engaged in action, there may be associations (S_2 && $B_{(i != 1)}$) which by virtue of containing S_2 appear to be both reliable and relevant, but which are, in fact, purely coincidental. This can lead to superstitious behavior [Reynolds75], and is a particular problem if, for whatever reason, the creature never responds to S_2 with B_1 during the learning process. In nature, animals do demonstrate superstitious behavior so it is clear that this is a real problem, but it is also true that they can learn to associate a novel stimuli with a known stimuli, without assuming an instrumental contigency. This is, after all, what classical conditioning is all about. Thus, the challenge for us is to develop a single architecture which allows the creature to learn both classical (the behavior isn't important) and instrumental contingencies (the behavior is important), and if possible, avoid behaving superstitiously.

We will now turn to a discussion of our actual model.

5.5.2 Changes in Internal Variables Drive the Learning Process

In our model, outcomes generally correspond to significant changes in the level of an Internal Variable. More properly, the Internal Variable should be thought of as the ultimate outcome which drives the learning process. The animal can also learn intermediate outcomes as well, where the intermediate outcomes are simply "known" contexts which in turn are associated with the eventual change in a Internal Variable. As mentioned above, this is the basis for chaining together behaviors. However, ultimately the value of these secondary, conditioned or intermediate reinforcers (different words for the same thing) is derived from the change in the Internal Variable. One implication of this approach is that our system takes a distributed approach to learning. Each Internal Variable is responsible for its own learning. While in our system they all use the same approach, there is no reason why this should necessarily be so. Indeed, the evidence from ethology would suggest that different motivational systems learn in different ways, and attend to different features of the environment [Davey89].

Whenever an Internal Variable undergoes a significant change, the variable seeks to explain the change in hopes of discovering a new association which will make it easier for it to be satisfied in the future[22]. However, since ours is a value-based system, ultimately the variable has to assign a value to relevant associations. Intuition suggests that

22. Actually, the designer may specify whether they want a given Internal Variable to learn or not. Note that in the discussion that follows we intermix Internal Variable and motivational variable. A motivational variable is just of kind of Internal Variable.

this value should somehow reflect the time-discounted change in its value. To do this we use Sutton and Barto's temporal difference model of Pavlovian conditioning [Sutton90]. This model effectively forces alternative "explanations" (i.e., behaviors or object features) to compete to "explain" (i.e., make the best prediction of) the time-discounted changes in the value of the reinforcement variable (i.e., the outcome). Sutton and Barto's model is attractive because it is simple and yet explains most of the results of Pavlovian conditioning experiments. Note we use it as well to learn the value of instrumental contigencies as well, i.e. that the value of a particular conjunction of behavior and stimulus should reflects its time-discounted association with changes in the value of some motivational variable. The specific learning equation is shown in Figure 20

Figure 20 Learning Equation

$$\Delta V_{it} = \beta\left(\left[\lambda_t + \gamma\left[\sum_j V_{j(t-1)} \cdot A_{jt}\right] - \left[\sum_j V_{j(t-1)} \cdot A_{j(t-1)}\right]\right]\alpha\bar{A}_{it}\right)$$

where:

V_{it} = Max Value of RM i at time t

A_{it} = 1 if RM i passed find/filter and weight > .90, 0 otherwise

λ_t = Value of error variable at time t

λ_t = Value of error variable at time t

β = Learning Rate

α = Trace Rate

δ = Trace Decay Rate

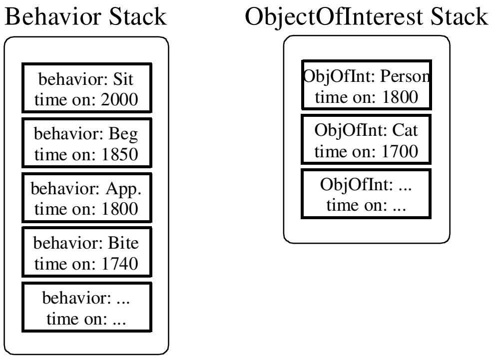
$\bar{A}_{it}$ = Trace for RMi at time t

Where the trace is defined as:

We follow the Sutton and Barto TD learning model in this paper. Thus, within a Discovery Group we use the following equation to learn the association between a Reinforcement Variable and one or more Releasing Mechanisms. The first equation specifies how the MaxValue associated with a given RM in the Discovery Group will change in response to the value of the Reinforcement Variable. The amount it will change is proportional to an error defined to be the feedback actually received plus the discounted prediction of future feedback by all the Releasing Mechanisms in the Discovery Group less the prediction of feedback by all the Releasing Mechanisms on the previous time step. In other words it changes by an amount proportional to the amount of feedback unexplained by the set of Releasing Mechanisms in the Discovery Group. In effect, they are competing for value, where the value is the time-discounted value of feedback. The second equation is used to specify a "stimulus" or memory trace for a given Releasing Mechanism. It basically does temporal averaging of the state of the Releasing Mechanism, where A is either 1 or 0 (i.e. active or in-active) and A bar ranges between 0 and 1. This term is used by Sutton and Barto to explain temporally discontiguous associations between a CS and a UCS (e.g. when the CS turns off prior to the start of the UCS).

The fundamental insight that allows us to integrate this learning algorithm into our existing action-selection algorithm is that the "value" to be learned for a stimulus is represented by the MaxValue associated with the Releasing Mechanism sensitive to that stimulus. Recall that Releasing Mechanisms return a value between some minimum and maximum, with the maximum value occurring when the Releasing Mechanism's object of interest appears at some optimal distance from the creature. As pointed out in 5.1.1

Figure 21 FIFO Memory Model



Memory is organized as 2 FIFO queues. The Behavior stack (shown on the left) keeps track of the last N behaviors performed, and the ObjectOfInterest stack (shown on the right) keeps track of the last N objects of Interest, where the objectOfInterest is the object which was the focus of the winning behavior on a given tick. In either case, an item is only added to the stack when it is different than the current front of the stack. It is from this stack of behaviors and objects from which the motivational variables select in order to learn potentially useful associations.

the value of a Releasing Mechanism should reflect the usefulness of its associated Behavior to the creature, and hence its maximum should be equal to the time-discounted usefulness of its object of interest (which the Behavior will exploit).

In all cases, then, we are learning the appropriate MaxValue to be associated with a given Releasing Mechanism. But on which conjunctions of behaviors and stimuli should the motivational variable focus in order to explain a change in its value? The answer is that the variable looks at the creature's recent behaviors and at the recently changed features of (nearby) objects of interest, to see if any of these may be associated with the outcome. Essentially, it says “let's check short term memory and start watching the last n behaviors and objects of interest to see if they continue to be associated with changes in this motivational variable from now on.” This focus on events which are in close temporal proximity to the change in the motivational variable follows from our earlier discussion of the importance of immediate feedback in animal training.

5.5.3 Short-Term Memory

Specifically, the Behavior System maintains two FIFO memories that are essential for learning. One memory keeps track of the last N unique behaviors that have been active, and another keeps track of the last N unique objects of interest (see Figure 17). The active behavior on a time step is the leaf behavior which ultimately won during action-selection. The object of interest is the pronome associated with that behavior. Note that in both cases the memory is based on unique instances and not time. Thus, a behavior or ObjectOfInterest only gets pushed on its respective memory stack if it is different than the behavior or ObjectOfInterest already on the top of the stack. This idea is taken from Killeen: “Decay of short-term memory is very slow in an undisturbed environment and dismayingly fast when other events are interpolated... Such distractors do not so much subvert attention while time elapses but rather by entering memory they move time along” [Killeen 94]. The intuition here is that is that it is the number of events that matter and not the space of time over which they occur. Indeed he presents evidence which he says shows how the results of traditional operant conditioning experiments can be explained using this memory model.

Foner and Maes [Foner94] also emphasize the importance of contiguity in learning. However, because short-term memory in our system is based on unique behaviors or objects of interest, our focus of attention is not strictly equivalent to temporal or spatial contiguity.

This is a very simple memory model. In the future, we will probably move to a system in which each motivational variable has its own memory-list of potentially significant objects or events, again stripped of their temporal information. For example, a motivational variable such as hunger might copy the state of the memory stacks when it undergoes a significant change (e.g. when it eats a mushroom) so that subsequently if something important happens (e.g. the creature gets violently ill), it can assign blame (e.g. no more mushrooms). This would then make it possible to model long-time-span learning phenomena such as taste aversion.

Thus, the motivational variable looks to short term memory for the behaviors and objects that the creature has performed or encountered recently. It then needs to construct Releasing Mechanisms to track possible conjunctions of stimuli and behaviors and via the Learning Equation have those Releasing Mechanisms compete to explain the change in the motivational variable's value. The details of this process are the topics of the next several sections.

5.5.4 BehaviorStimulus Releasing Mechanisms

We have defined a new type of releasing mechanism called a BSRM or BehaviorStimulus Releasing Mechanism to track conjunctions of behaviors and stimuli. Like typical Releasing Mechanisms, a BSRM detects when a certain condition is true and returns a value, but in this case the condition is the conjunction of the state of a behavior and the state of a given external stimuli. Whenever, a motivational variable wants to discover if there is an instrumental contingency between performing a behavior B in context S and a subsequent change in its value, it creates a pair of linked BSRMs. The first fires on (B && S), or when both the behavior is active and the stimulus is present. The second fires on ((!B) && S), or when the behavior isn't active but the stimulus is present. For example, if "hunger" wanted to discover whether there was a contingency between "Sitting" and hearing "SIT", it would create a pair of BSRMs, one which fired on ("Sitting" && "SIT"), and another which fired on (("Not Sitting") && "SIT"). As mentioned earlier, by looking at the relative reliability of the two BSRMs we can detect if there is an instrumental or classical contingency.

There is an important assumption built into the logic of the (B && S) BSRM, namely that the stimuli must turn on prior to, or within a few ticks of the behavior becoming active. The idea here is simple, here you are only interested in stimuli which identify a context in which to perform a given behavior, you are not interested in stimuli which may be the product of performing the behavior (the system does learn this but as a by product of the learning equation).

Thus, the linked pairs of BSRMs are the fundamental units which are used to learn useful conjunctions of stimulus and behavior. Ultimately, the BSRMs constructed by a single motivational variable compete to explain the change in the value of the variable, and as they compete via the learning equation, the system adjusts the MaxValue of the respective BSRMs to reflect the "learned value". This is done, as described below, within the context of a structure called a Discovery Group.

5.5.5 Discovery Groups

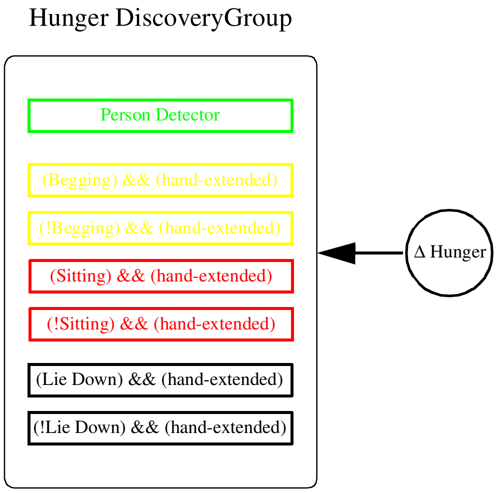
A Discovery Group is simply a collection of Releasing Mechanisms (mostly BSRMs) which are created by the motivational variable in order to discover useful instrumental and classical contigencies associated with significant changes in the value of the motivational variable. Each motivational variable has its own Discovery Group.

Discovery Groups are populated in the following manner. When a motivational variable undergoes a significant change, it goes to short-term memory and for each of the top M objects on the ObjectOfInterest stack, it checks if its Discovery Group already contains a Releasing Mechanism which fires on the presence of that class of object, and if not, it creates one and adds it to the Discovery Group. The purpose of these Releasing Mechanisms is to learn the "value" associated with their respective Objects Of Interest. These Releasing Mechanisms also serve an important purpose by detecting changes in the accessible features of their respective Objects Of Interest.[23] Subsequently, when the motivational variable undergoes a change, it queries each of these Releasing Mechanisms and for each field that has changed, it goes through the top N behaviors on the behavior stack, and creates (assuming one doesn't already exist), a BSRM pair for that stimulus and behavior. An example of a populated Discovery Group is shown in Figure 22 Thus, if there are M behaviors and N features which have changed, a total of MxNx2 BSRMs are created and added to the Discovery Group, assuming none exist already (i.e. only unique BSRMs are added to the DG).

It is important to note that since there is one global memory stack for the creature, the Behaviors and Object Of Interest from which the BSRMs are created may initially be derived from of any of the initial subsystems of the Behavior System. There are several implications here. First, it means that the system does not learn new Behaviors or to pay attention to new types of objects (i.e. for an object to appear on the memory stack, it must have been the pronome for some Behavior). Rather it learns how to use these Behaviors and Object Of Interests in new contexts so as to satisfy new motivational goals. Ethologists might defend the first point (re-use existing behaviors), but the second constraint (only pay attention to objects of interest which you already know about) is, at best, a simplification. Second, most animals can not learn to re-use any innate behavior in any arbitrary context. As Lorenz so aptly puts it: "There are no rat prostitutes" [Lorenz73] reflecting the fact that it is all but impossible to train a rat to perform a sexual display for some reinforcement other than a "willing partner".

23. In Hamsterdam this is done using a bit of Inventor magic. Inventor provides the ability to query an instance of a class at run-time and get a list of its fields, their names and types, and their values. We use a convention that fields which represent features of a creature which are meant to be visible to other creatures are of a particular types. Thus, the change detectors described above query their object of interest for the list of fields of this particular type and then subsequently watch the state of the fields. It is assumed that creatures (and their creators) are honest and update the fields to reflect their true state.

Figure 22 Discovery Groups (DG) are built and used by motivational variables to explain significant changes in their value



Discovery Groups (DG) are built and used by Reinforcement Variables (e.g. motivational variables) to explain significant changes in their value. For example, when hunger undergoes a significant change, it looks at short-term memory for the most recent behaviors performed and objects of interest encountered and adds the appropriate RMs and BSRM pairs. In the example above, there is one RM which fires on the presence of a person, and 3 pairs of BSRMs which fire on particular conjunctions of the state of a behavior and a stimulus. The Motivational Variable chooses which BSRMs to create and add to the Discovery Group based on the contents of the memory stacks. See text for details. Within a DG, the Barto-Sutton TD learning algorithm is used to discover the appropriate MaxValue for each RM.

Once they are added to the Discovery Group the BSRMs compete, using the Sutton-Barto model to “explain” the change in value of the motivational variable, and as they learn their association with the motivational variable, they change their maximum value accordingly. Thus, a BSRM that has no apparent association with the state of the motivational variable will wind up with a maximum value close to zero. By contrast, a BSRM which has a strong association will result in a maximum value that approaches the time-discounted value of the motivational variable.

5.5.6 How New Appetitive Strategies are Integrated into the Behavior System

In addition to using the learning equation to discover the appropriate values for the BSRMs in the Discovery Group, the system also keeps track of the reliability of each BSRM using the metric described in Figure 23 below. The Reliability metric is used to measure whether there appears to be an instrumental contingency. If the reliability of (S && B) is in fact higher than (S && (!B)), and if the learned MaxValue of the (S && B) is above a certain threshold, then the motivational variable adds the behavior to its repertoire of appetitive behaviors. Specifically, it makes a copy of the behavior B, and creates a new Releasing Mechanism which fires on stimulus S, and has a MaxValue equal to the MaxValue of the (B && S) BSRM. This Releasing Mechanism is then added to Behavior B’s list of Releasing Mechanisms. Behavior B is then added to the Motivational Variable’s top level Behavior Group. This Behavior Group is typically composed of one consummatory behavior and multiple appetitive behaviors, and the effect of this

Figure 23 Reliability Metric

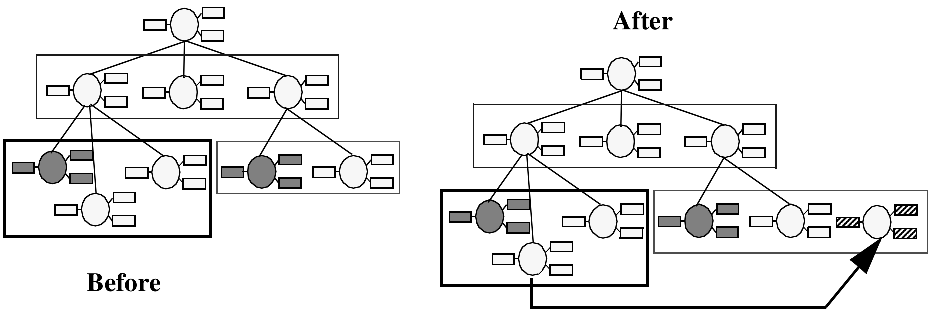
$$\text{Reliability}_{(B \wedge S)} = \frac{\#(\text{BehaviorOn} \wedge \text{StimulusOn} \wedge \text{Reinforcement})}{\#(\text{BehaviorOn} \wedge \text{StimulusOn})}$$

$$\text{Reliability}_{((\neg B) \wedge S)} = \frac{\#(\text{BehaviorOff} \wedge \text{StimulusOn} \wedge \text{Reinforcement})}{\#(\text{BehaviorOff} \wedge \text{StimulusOn})}$$

The system keeps track of reliability for each member of a BSRM pair. An instrumental contingency is assumed only if the reliability when the behavior is on (the first equation) is higher than when the behavior is off (the second equation). Note, that "on" means having a memory trace which is greater than some epsilon (see the learning equation). That is, the actual Releasing Mechanism may be off, but its trace is still high enough to be considered "on" for the purposes of learning.

addition is to add yet another appetitive behavior. More concretely, in the case of sitting (A) on command (S) for food (O), the sitting behavior and the set of releasing mechanisms (e.g., an outstretched hand) that indicate it's time for Silas to do his trick will be added to the Behavior Group that is principally concerned with feeding. Example of this are illustrated in Figure 24 and Figure 25 Subsequently, we will refer to this process as "expanding the BSRM".

Figure 24 Strategy: Learn to re-use existing behaviors in new contexts



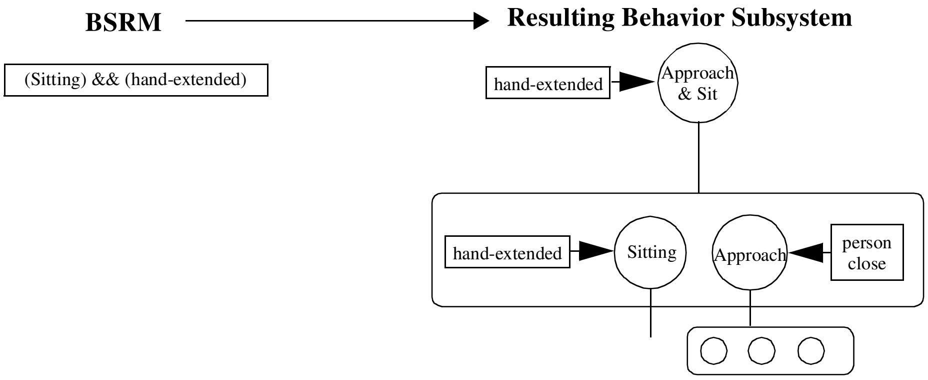
Ethologists such as Lorenz view animal training, and operant conditioning in general as a process of discovering that an existing behavior, when performed in a novel context for that behavior, may reliably lead to the satisfaction of a previously unassociated motivational variable. As a result, the behavior is "copied" (together with its new releasing mechanisms) into the primary behavior system associated with the motivational variable.

Even after the BSRM has been expanded and the appropriate Behaviors and Releasing Mechanisms have been added into the behavior system, the learning mechanism continues to use the BSRM to learn the appropriate value for the RM associated with the new behavior. As the BSRM changes value in response to further learning, it changes the MaxValue of the newly installed Releasing Mechanism. In addition, the BSRM fires if either the original behavior or its copy is active.

5.5.7 How Does the System Handle Chaining

Chaining of stimulus and behavior pairs, i.e. ((S_2 && B_2)->S_1) && B_1)->R, is a by-product of using the Sutton-Barto learning equation[24]. For example, as the trainer trains the first pair (S_1&& B_1)->R, the maxValue of the (S_1&& B_1) BSRM will grow in proportion to the reinforcement received. After a number of trials the trainer can then move

Figure 25 How a BSRM is expanded into a Behavior Subsystem



This shows an example of the type of behavior subsystem which is created from a BSRM when the motivational variable decides that it represents a useful new appetitive strategy. Note that a generic approach behavior is added so that the creature will first approach the object of interest (in this case the person) and then perform the action. The use of Pronomes makes this possible.

on and begin training the next pair ((S_2 && B_2)->S_1). In this case, the reinforcement is simply the presence of S_1 (S_1 is acting as a conditioned reinforcer). If the creature has learned the first lesson, it will respond with B_1 and the trainer will presumably give reinforcement R. Now even though there is no temporal overlap between (S_2 && B_2) and R (i.e. the memory trace $\overline{A}$ of the (S_2 && B_2) BSRM is zero by the time R is received), there is a temporal overlap with (S_1&& B_1) which in effect predicts the eventual reinforcement R. By virtue of the learning equation, the maxValue of the (S_2 && B_2) BSRM will grow in proportion to the reinforcement predicted by the (S_1&& B_1) BSRM, i.e. in proportion to the MaxValue of the (S_1&& B_1) BSRM. In this way the system accommodates chaining of stimulus and behavior pairs.

5.5.8 Detecting Classical Contingencies

The system as described will discover useful associations between stimuli and behaviors, as long as they overlap in time and that they eventually lead to reinforcement. As mentioned earlier, in many cases the value of a stimuli comes not from setting the context for a particular behavior, but rather because it is predictive of another stimuli which does have a useful association with an instrumental behavior.[25] For example, suppose stimulus S_y follows stimulus S_x regardless of what the creature does. While the creature may not be able to affect this pattern, it may want to take advantage of it, particularly if S_y is associated with a behavior B which does have an instrumental contingency (i.e. leads to reinforcement). In this case, it might pay to associate S_x with B (i.e. add a Releasing Mechanism which fires on the presence of S_x to B's set of Releasing Mechanisms) since this might allow the creature to get more reinforcement or get it sooner. Indeed, S_x might be such a good predictor of S_y that it is no longer important to pay

24. The subscript refers to the order in which the behavior and stimulus is trained.

attention to S_y. So part of the learning process must be to discover stimuli which aren't currently associated with instrumental behaviors but which appear to have value as predictors of the state of other stimuli which are. Given such "predictive" stimuli, we want to build Releasing Mechanisms which fire on their presence and then add them to the appropriate instrumental behaviors.

The problem in a nutshell is that BSRMs only track conjunctions of behaviors and stimuli which overlap in time, and we want to use them to discover associations between behaviors and stimuli which may not overlap in time. However, by comparing the relative reliability of the (S && B) and (S && (!B)) members of a BSRM pair we can easily detect if there is evidence that (a) a given stimulus does have predictive value, but (b) should not be associated with any of the behaviors with which it currently overlaps. For example, consider the case of a single stimulus S_x. If for all (S_x && (B_i)) and (S_x && (!B_i)) pairs in a Discovery Group it is true that the reliability of the (S_x && (!B_i)) BSRM is higher than its (S_x && (B_i)) counterpart, and if it is also true that the MaxValue of at least one of the (S_x && (!B_i)) BSRMs is above a threshold, then the system can provisionally conclude that S_x has predictive value, but should not be associated with a behavior with which it currently overlaps. If it didn't have incremental predictive value the MaxValues would be low. Similarly, if it should be associated with a behavior with which it currently overlaps, then one would expect to see that behavior's (B && S) BSRM have a higher reliability than its (S_x && (!B_i)) counterpart. In any event, if the system concludes that S_x has predictive value, it assumes that it is predictive of a stimulus which changes state after S_x and which is already associated with an "known" instrumental behavior. That is, the system finds a stimulus S_y which both changes state in close temporal proximity to the stimulus in question (i.e S_x) and which is already known to have an instrumental contigency (i.e. is a member of a BSRM which has already been expanded) with a Behavior B. It then builds a new Releasing Mechanism which fires on S_x and add it to B's set of Releasing Mechanisms (which of course already includes one which fires on S_y).

In detail the process works as follows. First, we query the Releasing Mechanism which monitors changes in the ObjectOfInterest's features and find the feature S_y which both

25. In the preceding discussion, the assumption is that it is the conjunction of a stimulus and a behavior which is instrumental to gaining reinforcement. That is, the value of a stimulus is derived from the fact that it identifies a context in which performing a particular behavior will lead to reinforcement. Thus, a BSRM is defined to fire on conjunctions of stimuli and behavior. This is a fine model in the case of animal training since successful trainers insure that there is appropriate overlap. Here we want to discover useful associations between stimuli and behaviors, where there is no overlap, and where the stimuli may coincidentally overlap with irrelevant behaviors. This is a fundamental problem in systems like ours in which the creature does not behave randomly. In this kind of system looking at simple conjunctions of behavior and stimulus will not discover useful associations between stimulus and behaviors which don't overlap. The whole purpose of the mechanism described in this section is to identify these potentially useful associations. A useful association, in this context, is one in which a given stimulus predicts the state of another stimulus which is already associated with the behavior. By recognizing these kinds of associations the creature can potentially get reinforcement more quickly and avoid superstitious behavior.

changes state next after S_x and which is also associated with an expanded BSRM. From the (S_y && B) BSRM we then get the behavior B to which eventually will be added the new S_x Releasing Mechanism. We then find the (S_x && B) BSRM and set its maxValue and reliability to be equal to the maximum values found in the set of (S_x && (!B_i)) BSRMs in the Discovery Group. The effect of this is that on the next time step, the (S_x && B) BSRM will be expanded yielding the equivalent of ((S_x || S_y) -> B), that is, Behavior B will have two Releasing Mechanisms, one that fires on S_x and the other on S_y. This will cause the Behavior B to subsequently respond when either S_x or S_y are active, and through the learning process this will have the effect of tuning the MaxValues of the S_x and S_y RMs to reflect their appropriate values. Note, that if S_x does in fact have incremental predictive value (i.e. it predicts the change in S_y), the dynamics of the learning equation are such that S_y may actually drop in value in response.

5.5.9 Learning the Time Course

As mentioned earlier, in a world of uncertain rewards an animal needs to learn not just new appetitive behaviors, but also how long to engage in a given appetitive behavior before giving up and trying something else. We use a very simple approach to this problem: as part of the learning process, the Behavior incorporates its expected time to feedback and the associated variance. The Level of Interest for the Behavior is then held constant (and high) over the expected time to reward, and from that point on is allowed to decay linearly to zero over a period of time.

5.5.10 A Variable Learning Rate

It is also important to learn the appropriate learning rate. It is a well-known phenomena that animals taught under a variable reward rate (for example, they are rewarded only every third time), will take longer to learn a given lesson, but will correspondingly take longer to forget the lesson under extinction. Similarly, animals will reacquire associations that have been extinguished faster than they will acquire new associations. Both phenomena are consistent with the idea that the learning rate is somehow correlated with the reliability of feedback. Bees appear to use different learning rates for learning the odor, color and shape of food sources. Once again the rates seem to be correlated with the reliability of the stimuli [Gould 94]. Neither of these observations would be predicted by the standard fixed-rate learning equation. Indeed, researchers such as Gallistel and Cheng [Gallistel90, Cheng95] base much of their criticism of associative learning on just these reasons.

To address these observation, we propose that the learning rate should be proportional to some measure of the observed reliability of receiving, or not receiving, feedback. If the reliability is high -- e.g., a behavior is always rewarded, or always not rewarded -- then it makes sense to have a high learning rate, i.e. to pay attention to each observation. On the other hand, if the observed reliability is low --e.g., a behavior is sometimes rewarded, and sometimes not --then the learning rate should be low, reflecting the uncertainty associated with any one observation.

In our model we base the learning rate on a moving average of what we call the “Reliability Contrast” as described in Figure 26 The Reliability Contrast (RC) is the differ-

Figure 26

Reliability Contrast and the Learning Rate

Reliability Contrast

$$RC = \frac{\#(\text{active \& feedback})}{\#(\text{active})} - \frac{\#(\text{active \& no feedback})}{\#(\text{active})}$$

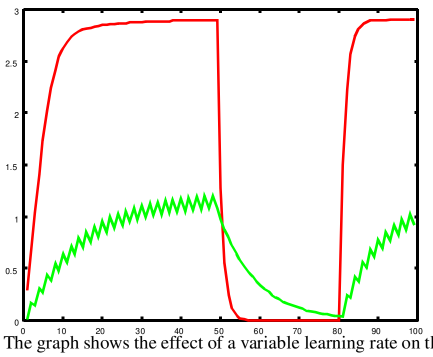
Learning rate:

ence between the observed likelihood of receiving feedback at least once while the behavior is active and the observed likelihood of receiving no feedback while the behavior is active. Other researchers have used similar measures of reliability (see Maes90b in particular).

This variable learning rate allows us to accommodate one of the two weaknesses of the learning equation, namely the partial reinforcement effect. It will also accommodate the phenomena of rapid re-acquisition after extinction because in extinction the learning rate is high, and so it responds to the change rapidly. Both effects are shown below in Figure 27 It does not, however, accommodate the phenomena of prior exposure reduc-

Figure 27

Variable Learning Rate and Re-acquisition



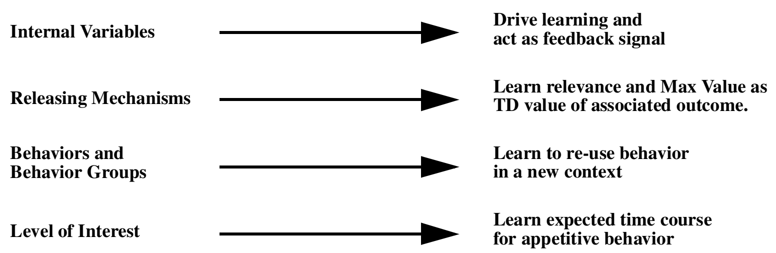
The graph shows the effect of a variable learning rate on the time-course in the associative strength of a stimulus under acquisition (t<50), extinction (50<t<80), and reacquisition (t>80) in 2 feedback ratio scenarios ((1) = rewarded each time, and (2) = rewarded every other time). The learning rate is calculated in each case on a moving average of the reliability contrast. Since the reliability contrast is lower in the case of the variable reward scenario (2), its learning rate is lower and thus its decay in extinction is more gradual.

ing the learning rate. We also do not yet model the transition from a learned response to essentially a habit (i.e. when the learning rate decays to 0 and learning stops).

5.5.11 Summary

As should be clear, our approach to learning is closely integrated into our action selection architecture. This is summarized in Figure 28. We have taken a standard variant of TD learning, embedded it in our ethologically inspired architecture, and then used it to do learning in an ethologically plausible manner.

Figure 28 Role of action-selection structures in learning process



5.6 Integrating External Control

Motivational and behavioral control is accomplished by adjusting the factors which influence the relative strength of a Behavior, or group of Behaviors. This may be done in a number of ways.

- First, motivational control is provided via named access to the Internal Variables which represent the motivations or goals of the Behavior System. By adjusting the value of a given motivational variable at run time, one can make the creature more or less likely to engage in Behaviors which depend on that variable.
- Second, the constituent parts of a Behavior are also accessible, and this provides another mechanism for exerting behavioral control. Releasing Mechanisms (see Figure 7) act as “object-of-interest” detectors for a Behavior. By modifying the kind of “object-of-interest” to which a given Releasing Mechanism is sensitive, one is in effect modifying the conditions under which a given Behavior will become active. For example, the “marking” Behavior of a dog may rely on a Releasing Mechanism which triggers on “fire hydrants”. By modifying it at run-time so that it triggers on “user’s pants leg”, the resulting behavior of the dog will be very different. One can also adjust the “strength” of a Releasing Mechanism so as to raise or lower its effect on the value of the Behavior (and thus the likelihood of the Behavior becoming active when the Releasing Mechanism is triggered).
- Third, Level of Interest provides a natural way of providing proscriptive control because it adjusts the effective likelihood of performing a given behavior independent of the intrinsic strength of the behavior (i.e. the combined strength of the relevant internal and external factors.
- Fourth, the Behavior System is structured so that action-selection (i.e. the process of choosing which Behavior should be active at any given instant), can be initiated at any node in the system. This allows an external entity to force execution of a partic-

ular part or branch of the Behavior System, regardless of motivational and sensory factors which might otherwise favor execution of other parts of it. Since branches often correspond to task-level collections of Behaviors, this provides a way of achieving task-level control.

Since each creature has its own sensory system, it is very straightforward to provide "imaginary" sensory input. For example, objects may be added to the world which are only visible only to a specific creature. The presence of these objects, however, may trigger certain behaviors on the part of the creature. It should be noted that this may be done more directly by manipulating the creature's Releasing Mechanisms. However, the advantage of this technique is that it does not require the external entity to know anything about the internal structure of the Behavior System.

The mechanisms described above for controlling the Behavior System naturally support Maes' second dimension of control. For example, by adjusting the level of a motivational variable which drives a given branch of the Behavior System, the director is expressing a weighted preference for or against the execution of that behavior or group of behaviors.

The multiple imperative forms supported by the Motor Controller allow the director to express weighted preferences directly at the motor level. For example, at one extreme, the external entity may "shut off" the Behavior system and issue motor commands directly to the creature. Alternatively, the Behavior system can be running, and the external entity may issue persistent secondary or meta-commands which have the effect of modifying or augmenting the output of the Behavior system. For example, the external entity might issue a persistent secondary command to "wag tail". Unless this was explicitly over-ruled by a Behavior in the Behavior system, this would result in the Dog wagging its tail. Or, for example, the external entity might suggest, via a meta-command, that if the dog chooses to move, it should move with a "trot", or "use any gait but a trot". Meta-commands may also take the form of spatial potential field maps which can be combined with potential field maps generated from sensory data to effectively attract or repel the creature from parts of its environment.

See [Galyean 95] for an example of how these multiple levels of control were used to control an animated dog within the context of an interactive narrative.

The key point to note here is that in this architecture it is not necessary to add any new functionality (other than run-time access to the behavior and motor systems) in order to support multiple levels of external direction. In fact, many of the very same features which are necessary for robust autonomous behavior (e.g. motivational variables or multiple imperative forms for motor commands) turn out to be essential for the integration of external control.

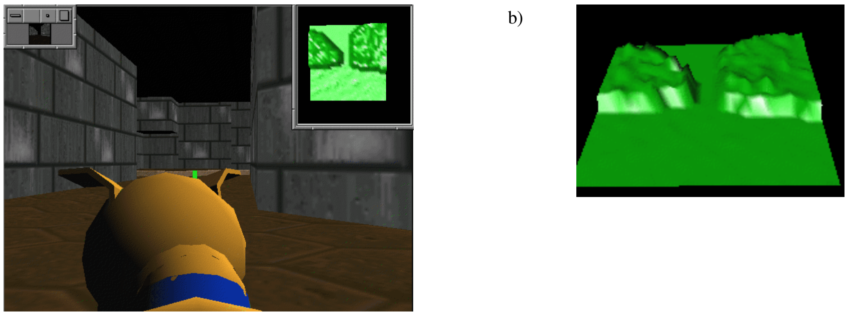
5.7 Sensing and Synthetic Vision

The preceding discussion has assumed the creature has some means of sensing its environment, but has not specified how that is to be done. Here we will describe our approach to sensing and, in particular, our use of synthetic vision for low-level navigation and obstacle avoidance. In the course of this discussion we will see how we have taken ideas from ethology as to the clues bees may use for obstacle avoidance (i.e. motion energy) and incorporated them into our computational model. We will also see how some of the ideas in the previous section such as Level Of Interest have practical application.

There are at least three forms of sensing available to creatures in our system:.

Figure 29 Synthetic Vision is used for obstacle avoidance and low-level navigation

a) b)



Silas uses motion energy and other features recovered from his "vision" sensor to perform obstacle avoidance and low level navigation. In (a) we see the scene from a camera mounted on Silas's back. The image in the window in the upper left is the actual (32x32) image rendered by the "vision" sensor using an Inventor perspective camera. The window in the upper right shows the recovered "motion energy map" (see text). In (b) the "motion energy map" is displayed from a low angle to give a better sense of its structure.

- Real-world sensing using input from physical sensors.
- "Direct" sensing via direct interrogation of other virtual creatures and objects.
- "Synthetic vision" using computer vision techniques to extract useful information from an image rendered from the creature's viewpoint.

In the ALIVE interactive virtual reality system [Maes96], creatures rely on all three forms of sensing. For example, the user's proxy creature in the virtual world uses real-world sensing to recover the user's position, posture and gestures, and sets state variables which are accessible to other creatures via direct sensing. Thus, if "Silas", the virtual dog in the ALIVE system, wants to know if the user's hand is extended, he simply retrieves the value of the appropriate state variable from the proxy creature. While this approach is very simple and fast, it has a number of limitations. First, it requires agreement on a common protocol with which to make and respond to inquiries. Second, direct sensing does not help with the problem of obstacle avoidance and low-level navigation (note: by low-level navigation we mean the capability to reactively move to a goal, possibly making detours as necessary, but without possessing global knowledge, other than the location of the goal, and without detailed path planning).

A number of researchers have suggested using computer vision techniques to address the navigation and obstacle avoidance problem in computer graphics. Among them are Reynolds [Reynolds87], Renault [Renault90], and Terzopoulos [Tu94]. There are a number of motivations for doing so:

- First, it may be the simplest and fastest way to extract useful information from the environment (e.g. using vision to steer down a corridor as opposed to using an analytical solution). This may be particularly true if the system can take advantage of the underlying hardware.
- Second, synthetic vision may scale better than other techniques in complex environments.
- Third, this approach makes the creature less dependent on the underlying representation/implementation of its environment because it does not rely on other creatures and objects to respond to particular queries. Nor does it require knowledge of the data structure used to implement its world. All it needs to be able to do is render the world.

Here we discuss two alternate approaches to using synthetic vision to guide low level navigation and obstacle avoidance. In the first approach, the "potential field approach" we use the recovered image to build a potential field representation of the creature's immediate surroundings and use the resulting gradient to arrive at a bearing to steer. In the second approach, the "motion energy approach", we use a very approximate measure of motion energy combined with the extraction of a few simple features to guide navigation. In particular, we will show how this latter approach, when combined with a reactive behavior system is able to generate robust behavior. While other authors have suggested the usefulness of optical flow for guiding movement in robots (in particular see [Duchon96]), our contribution is (a) to show how a cheap first order approximation of motion energy may suffice, and (b) to stress the usefulness of the approach as a general technique for performing obstacle avoidance and navigation in autonomous animated creatures.

5.7.1 Direct Sensing

By direct sensing we mean the ability of a creature to directly interrogate another creature for key information, such as the creature's position, or some aspect of its state. While highly unrealistic, this approach is fast and very easy to implement. All that is required is a common protocol. In the Hamsterdam system this is implemented in the following manner. All creatures respond to a core set of inquiries, for example, "getPosition()" returns the position of the creature, and "getHeading()" returns the heading. In addition, through the run-time support provided by Inventor, it is possible to named access to any of a creature's fields. For example, in ALIVE, the person's proxy creature in the ALIVE world has boolean fields for each of the person's possible gestures and postures (e.g. the "extendedLeft" field is set true if the user's hand is extended to the left). Thus, if the Dog wants to "know" if the user is performing a given gesture, it simply checks the state of the appropriate field.

Creatures in Hamsterdam typically have one or more "sniff" sensors with which to perform direct sensing. The sniff sensor is intended to be an idealized "nose", which given a list of the kinds of creatures for which to sniff, will find the closest instance of each kind of creature on a given tick. Specifically, on each update cycle it will cache a pointer to the closest instance together with other information such as the creature's world position and its bearing and range relative to the sensor. This information is used by the Releasing Mechanisms of the creature in a number of ways.The simplest type of Releasing Mechanism (called a CreatureRM) simply detects the presence of another creature, and thus needs no other information aside from the distance and bearing. A more complicated type of RM (called a CreatureFieldRM) uses the cached pointer to the creature to check the state of one or more specific fields in the creature.CreatureFieldRMs can be written to apply logical operators such as AND (e.g. “extendedLeft” AND “personSitting”), OR (e.g. “extendedLeft” OR “extendedRight”), or NOT (e.g. NOT (“extendedLeft” OR “extendedRight”)) to arrive at the specific conditions under which they will fire.

Since sniff sensors automatically calculate the range and bearing to another creature relative to the sensor’s coordinate system, it is often convenient to use multiple sniff sensors placed at different parts of the body, for example, in each hand. This makes it easy for the creature to answer the questions such as: “is the object of interest closer to the left paw or the right paw”. However, this is for convenience only since this question could be answered using a single sniff sensor and by knowing the position of the respective paws.

The greatest problem associated with direct sensing, other than the need for a shared protocol is that it can provide the creature with super-natural sensing and this in turn can lead to super-natural behavior. For example, suppose the dog’s view of the person is blocked by an obstacle, or the person is behind the dog. In either case, the dog should not be able to detect attributes of the person, and thus should not respond to gestures made by the user. However, with a naive implementation of direct sensing (which is what we have currently implemented) the dog will sense things that he shouldn’t be able to sense, and this ultimately leads to behavior which is less than believable. While some of these problems can be addressed in a more sophisticated implementation of direct sensing (e.g. only sensing the attributes of creatures which are in a hypothetical field of view), this will not solve the problem of sensing the attributes of creatures which are in the creature’s field of view but obscured by other objects. To solve this latter problem you need to implement some type of synthetic vision such as is described below.

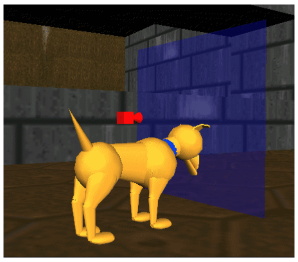
The fundamental problem is that direct sensing is “cheating” and it is easy to get caught in the lie, particularly as the environment and the interaction becomes more complex. For the behavior to be believable, the creature’s ability to sense its environment must also be believable.

5.7.2 Implementing Synthetic Vision

The fundamental idea is very simple: a creature may have one or more “vision” sensors which during their update phase render the scene using a graphics camera mounted at the position and orientation of the sensor. The resulting rgb image is extracted from the framebuffer and used as input into whatever perceptual mechanisms the sensor provides (e.g. build an egocentric potential field representation of the image or perhaps approximate the motion energy in the image). Typically, the vision sensor is at a fixed location relative to the creature’s body. For example, Silas has a single vision sensor which uses a perspective camera with a field of view of 90 degrees as shown in Figure 30 below.

To aid in visual tracking, false coloring is typically used when rendering the creature’s current object-of-interest. For example, if Silas’s current object of interest is the “food dish”, Silas’s vision sensor will, in effect, request that the “food dish” use a special color

Figure 30 Silas's vision sensor



Silas's vision sensor (the red camera above) is mounted above and behind his shoulders, but in the frame of his shoulders so it always points in the direction of movement. The field of view is 90 degrees and is indicated by the transparent blue rectangle above.

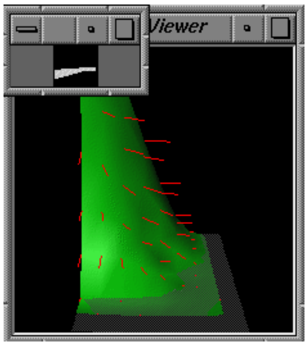
when rendering itself for the purposes of the vision sensor's render. In the subsequent analysis of the resulting image, it is easy for the system to detect if some portion of the object (e.g. the "food dish") is in its visual field (i.e. look for pixels with that color). This use of false coloring is analogous to Chapman or Whitehead's use of markers [Chapman91, Whitehead92]. This use of markers is particularly useful for answering a very common and important question, informally "how close am I to a given object" and formally, "what is the distance from the origin of the sensor to the closest point on the surface of the object in question". In our system we approximate this distance by finding the lowest "marker" pixel in the frame, assuming it is on the floor, and back-projecting the point into the camera's coordinate frame. While this approximation is only valid for objects which are not over-hanging and which rest on the floor, it appears to work well in practice, and in any event is a better approximation than taking the distance to the origin of the object (i.e. its center), or to a point of intersection on the object's bounding box.

We now turn to two alternative ways of using the image to guide low-level navigation and obstacle avoidance. In the first approach, egocentric potential fields, we use the image to build a potential field representation of the creature's immediate surroundings and use the resulting gradient to arrive at a bearing to steer. In the second approach, motion energy representation, we use a very approximate measure of motion energy combined with the extraction of a few simple features to guide navigation.

5.7.3 Egocentric Potential Fields

While not necessarily inspired by ethology, potential field or other gradient description representations have proven useful in autonomous creatures. Potential fields have been used extensively in robotics as partial task-based representations of the world [Arkin90, Payton90, Latombe91, Reynolds87]. It should also be noted that potential field methods have been used by researchers such as Giszter [Giszter94] to control short range movements. The underlying idea in this work is to represent some aspect of the world (or a task in the world, such as moving toward a goal) as a potential field with varying levels of potential. Typically, a gradient field is derived from the potential field, and that is used to arrive at a direction to move. The great advantage of potential field representations is that they are easily combined to arrive at a compromise solution. Arkin

Figure 31 Egocentric potential field generated using synthetic vision



[Arkin90] uses potential fields to implement a form of command fusion (say between “avoid obstacle” and “move-to-point”). This is an example of task-based representations being shared and combined among multiple behaviors. Payton views potential fields as a way of describing a plan [Payton90]. He refers to this as an “internalized plan”, that is a representation which allows “the raw results of search in any abstract state space to be made available for direct use within continuous real-time decision-making processes.” Thus, rather than providing a robot with an abstract route through an area, you provide it with a gradient field representation of the area. The robot of course can derive the abstract route from the gradient field, but more importantly, it can combine the map data with sensory data so as to react to “unexpected changes in the environment.”

In earlier versions of our system we took just this approach. We used synthetic vision to acquire an image of the scene from the vantage point of the creature. This was used to build an egocentric potential field map of the area within visual range of the creature. To do this we relied on a simple heuristic suggested by Horswill [Horswill93]: for each column in the image search up from the bottom to find the first “non-floor pixel”. Since the creature knew the position and orientation of the camera, and since usually the first non-floor pixel would be just above floor level, it was an easy matter to back-project the point into camera space and use the resulting point as the center of a potential charge which dissipated exponentially. This process was repeated for each column in the image. See Figure 31 for an example of potential field generated in this fashion. Once the field was computed, its gradient was calculated.

Borrowing an idea from Arkin [Arkin90], goals were represented as “schemas”, that is as a potential well that could be appropriately positioned with respect to the egocentric map derived from the vision system, and then combined to arrive at a composite map.

However, localized Potential Field techniques like this suffer from a number of problems:

- They tend to be too strong up close and too weak far away (this point was noted by Reynolds [Reynolds87] in his original discussion of vision techniques for animation). The result is that creatures who rely on such techniques for obstacle avoidance act as if they are near-sighted: they do not respond soon enough to the presence of obstacles and when they do, the movement tends to be very pronounced. Thus, the resulting behavior does not appear natural.

- It is not clear how best to scale and combine "schemas" with fields generated from sensory data. For example, how "deep" should the potential well representing the goal be? For example, if it is too deep it may overshadow areas of high potential corresponding to obstacles. On the other hand, if it is too shallow, it may not provide enough of a gradient to arrive at compromise solutions when combined with the potential field generated from sensory data.
- How are the gradients within a single potential field map combined? Ultimately, the creature needs only one bearing to steer, but within a potential field map there will be multiple gradients and the system needs some method to combine them. Does it use all of them and take an average, does it place more weight on the gradients closer to the center line, and so on?

5.7.4 The Motion Energy Approach

Recovering motion energy from the visual field and using it to perform obstacle avoidance is a simple, yet robust alternative to Potential Fields. This approach is inspired by research into bee navigation in which it appears that bees flying down a corridor will position themselves, relative to the walls, so as to balance the motion energy in their right and left eyes [Srivansan96]. Similarly, they appear to use the perceived motion energy to control their altitude and ground speed [Mura94]. Desert ants use the perceived motion energy from their ventral retinas as input into their path integration system which keeps track of their distance and bearing from the nest[Wehner96]. We have chosen to adopt a similar approach in our system. See [Duchon96] for an example of using optical flow to guide movement in a maze by a real robot.

Figure 32 Estimate of Approximate Motion Energy

$$FM_{ijt} = abs(fw \cdot (pixel_{ijt} - pixel_{ij(t-1)}) + (1 - fw) \cdot pixelOccupied(i, j))$$

where:

FM_{ijt} = Measure of energy at pixel(i,j) at time t

fw = flow weighting (between 0 and 1)

$pixel_{ijt}$ = rgb value of pixel (i,j)at time t

$pixelOccupied(i, j)$ = 1 if pixel non-black, 0 otherwise

We combine a very simple and approximate measure of motion energy with a measure of "mass" to arrive at a metric to guide low-level navigation.[26] First, we divide the image in half. Then for each pixel in each half we calculate the a measure of energy as shown in Figure 32.

26. Sumit Basu and I came up with this approach as a quick and dirty approximation to motion energy. I have subsequently re-implemented the approach using normal flow, rather than a simple difference. However, in informal tests I have not noticed any real difference in performance.

The first term corresponds to the difference between successive frames, and is intended to be a gross measure of the motion energy at that pixel. The second term of our motion energy equation is not a measure of motion energy at all, but rather is added to deal with the scaling problems associated with computer generated textures. Intuitively, it represents a binary measure of whether there is mass at that pixel. We use the coefficient *fw* to weight the respective contributions of the flow and mass.

Then for each half we sum across all of the pixels to arrive at the total motion energy in that half. The difference between the total motion energy in the right and left halves of the retina is used to determine the bearing to steer. For example, if the creature wishes to go down the center of a corridor, it should balance the total motion energy in each half: i.e. it should turn in the direction of the least motion energy. That is:

Figure 33 Control Law for Corridor Following

$$bearingToSteer = K \cdot \left(\sum_{left} FM_{ijt} - \sum_{right} FM_{ijt} \right)$$

where:

FM_{ijt} = Measure of energy at pixel(i,j) at time t

K = Gain

This simple control law will work in most cases for corridor following and obstacle avoidance. The notable exception is the case in which the creature is moving directly toward a wall such that the measure of motion energy is the same on the right and the left halves of the retina. In this case, the control law will tell the creature to steer straight, and the creature will eventually collide with the wall. One solution to this problem is to keep track of the total motion energy, and when the total motion energy is above some threshold and the energy is approximately the same on either side of the retina, then pick a direction to turn and begin the turn.

In fact, we choose to augment this basic control law by extracting more information from the image and using it in conjunction with a simple reactive behavior system to provide more robust low level navigation and obstacle avoidance. Specifically, using Horswill's approach [Horswill93] we keep track of the height of the first non-floor pixel in each column. We call the resulting vector the "ridge". The ridge is used to recover additional information, including:

- Openings - By using a measure of the slope (i.e. the difference in heights between $column_{(i)}$ and $column_{(i+d)}$ where *d* represents the desired interval) the system identifies openings and "left" and "right" corners (an opening is defined to be a sequence of a "left" and "right" corner). A "left corner" is indicated by a positive slope above a certain threshold, and a "right corner" by a negative slope whose magnitude is above the threshold.

- Free Paths - The ridge can be used to see if there is a free path to a point in its visual field. The point may correspond to a pixel belonging to the object-of-interest (detected as described above via false coloring), or it may be a point which is projected into the visual field (if the object is too small to be seen given the resolution of the retina). In either case, it tests to see if a straight line from the bottom center of the image to the point of interest intersects a non-floor pixel. If so, then the path is blocked. If not, then there is a possible free path to the object.
- Dangerously close obstacles - If elements of the ridge are near the bottom of the image (and hence very close to the creature), then the creature may be in danger of hitting an obstacle, particularly if the elements in question are in the middle of the image (and thus directly ahead). Thus, the system keeps track of the number of elements which are in a so-called danger zone. This is also a fail-safe in the case where the creature may be approaching a wall head-on and so the motion energy on the right and left is the same.
- Free Space - If the minimum of the ridge is close to the horizon line, then for all intents and purposes the creature is in free space.

In addition, to the “ridge” data described above we also keep track of the “target ridge” which is composed of the lowest pixel in each column which is associated with the object-of-interest (detected as described above through false coloring). This is used to answer questions such as: “is the object in my visual field”, “is there a free-path to the object”, and “how close am I to the object”.

5.7.5 Navigation and Obstacle Avoidance using the Motion Energy method and the Importance of Boredom

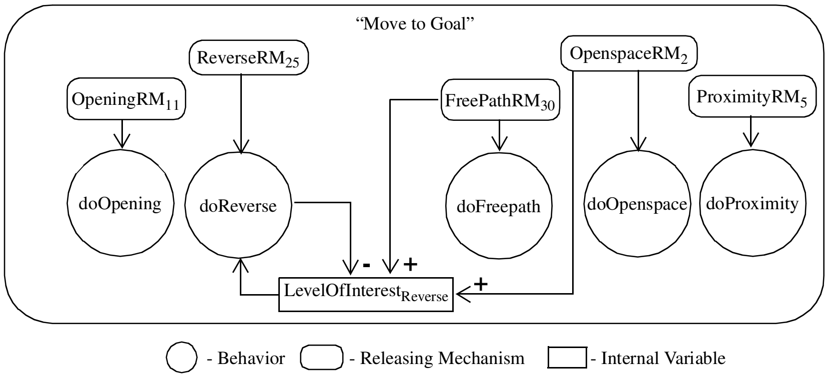
It is straightforward to design a behavior subsystem which uses this approach. First, one must create a number of releasing mechanisms, where each one fires on a particular visual event (e.g. one releasing mechanism might fire if an opening is detected). Second, one must design a number of behaviors, where once again each is targeted to respond appropriately to a given perceptual event. These behaviors are then made part of a behavior group which is, in effect, responsible for obstacle avoidance and low level navigation. An example of this is shown in Figure 34below. It is composed of five basic behaviors (“doFreepath”, “doOpening”, “doProximity”, “doReverse”, “doOpenspace”) each with a Releasing Mechanism which signals the appropriateness of the Behavior as indicated by the recovered features from the synthetic vision system. In addition to the synthetic vision (including false coloring of the current object of interest), the creature is endowed with a “nose” which gives them an egocentric bearing and direction to the current goal.

While most of the Behaviors are reasonably self-explanatory (e.g. “doOpening” moves the creature toward the middle of the perceived opening), one in particular, “doReverse” needs a bit more explanation. “doReverse” is intended for situations in which:

$$\mathrm{abs}(bearing_{goal}) \geq 1.57$$

That is, when the goal is behind the creature, the creature should orient toward the goal. This behavior, of course, makes perfect sense in an “open” environment where the creature is free to move in essentially a straight-line path to the goal. However, if the crea-

Figure 34 “Move to Goal” Behavior Group (Motion Energy Method)



This figure shows the behaviors which make up the generic “move-to-goal” behavior group of a creature which uses the Motion Energy method for obstacle avoidance and navigation. The MaxValue of the respective Releasing Mechanisms are shown as subscripts. Reverse is the only behavior with a variable Level Of Interest. When “doReverse” is active it reduces its Level Of Interest to avoid pathological circling. doReverse’s Level Of Interest stays low until there is a clear sign of progress, in this case that there is either a freePath to the goal or that the creature is in the open (as signalled by the Releasing Mechanisms which detect those conditions). This is an example of how Level Of Interest plays a useful role in avoiding pathological persistence.

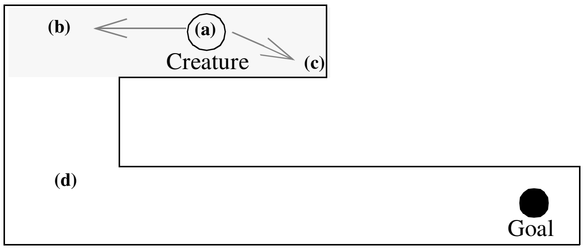
ture is in an environment in which it must make a substantial detour, then it may be necessary to travel for some period of time away from the goal (and hence have the goal behind the creature). For this to happen, “doReverse” must be inhibited from becoming active. This is done in our system by reducing the level of interest in “doReverse” (note: Level Of Interest is a measure of the Behavior’s interest in being active apart from any motivational or sensory input, i.e. it is used to model boredom, or lack of interest when it senses that it is not progressing toward its goal using a given strategy). However, once there is a clear indication of progress, as measured by either sensing a free path to the goal or that the creature is in open space, the Level Of Interest associated with “doReverse” rises to its normal level. The usefulness of this approach is illustrated in Figure 35

Using this approach, our virtual dog “Silas” is able to demonstrate robust obstacle avoidance and goal-finding (including situations in which the dog must take detours) in a number of environments including a world composed of the scenery from the “DOOM” video game. In our experience, the resulting behavior is more robust and more “natural” than that generated by our earlier implementation of the Potential Field approach. It is also interesting to note that the actual implementation was much simpler as well.

5.7.6 Limitations of the Motion Energy Approach

The current approach is surprisingly robust, but it is not without its limitations. First, it assumes that objects such as walls are rendered using textures (flat colored surfaces do not generate motion energy except at the edges). Second, the current implementation assumes that there is minimal texture on the floor. This is accomplished either by avoiding direct lighting of the floor, or by “inserting” a large constant colored polygon at floor level prior to rendering the scene. Third, it currently assumes a flat floor and will

Figure 35 When “doReverse” can lead to trouble



Reversing direction can be dangerous in a closed environment because it can cause the creature to get stuck. Consider the case of a creature at (a) which is facing (b), but wants to go to the marked “goal”. If it has a behavior which turns the creature in the direction of the goal when the goal is behind the creature (e.g. “doReverse”) it will tend to turn toward (c). But this will lead the creature into a corner, and even it turns away from the corner, “doReverse” will become active once it is out of the corner. Yet you do not want to remove “doReverse” from the creature’s repertoire or a creature at (d), facing (b) will take a very round-about path to get to the goal. The problem is that you want to suppress “doReverse” when the creature is making an extended detour. The behavior group in Figure 34 handles this problem by having “doReverse” lower its level of interest all the time it is active. This will make it less likely to become active in the future until its level of interest rises again. It will rise when there is some indication of progress (e.g. there is a freePath to the goal), or when it seems that “doReverse” won’t get the creature in trouble (e.g. the creature is in openSpace). This isn’t a perfect solution, but it does allow the creature to exhibit “detour” behavior without a global map. It also illustrates how Level of Interest can be used to solve particular problems.

not handle over-hanging obstacles. Lastly, since it is a reactive approach to navigation, and no path planning is done, the resulting path is unlikely to be optimal.

5.8 Implementation

Here we wish to give the reader a flavor for our approach rather than describe the full implementation of the Hamsterdam tool kit. The architecture described above is implemented as an object-oriented tool kit built on top of SGI’s Open Inventor. Specifically, the key components of a creature’s sensory system, behavior system and motor system are implemented as sub-classes of Inventor classes. There are a number of reasons for this approach:

- Open Inventor is an emerging standard for 3D graphics and this approach allows a creature designer to use industry standard modeling tools to define the geometry and materials for the creature. However, it is easy to subsequently integrate behavioral elements such as sensors into the models because as subclasses of Inventor classes they may inserted into the Inventor scene graph as appropriate.
- Open Inventor’s support of tree-structured networks of groups and nodes (this is the typical data structure for representing 3D graphics) is ideal for representing a tree-structured behavior system.
- Open Inventor defines a file format that is flexible enough to allow us to specify a creature’s behavior system in one text file and its geometry in another text file. This helps facilitate rapid turn around since re-compiling is not necessary.
- Open Inventor provides a very flexible run time environment which supports, among other things, named access to variables.has a powerful run time system as well as the ability to define default values for an object’s fields so that at run time one only need specify the values which are different from the defaults.

- Lastly and perhaps most intriguingly, because Inventor is the basis for VRML, it should be relatively straightforward to port the system to VRML. Sixth,

Thus, the Hamsterdam tool kit defines a set of classes (e.g. Behavior, Releasing Mechanism, InternalVariable, Behavior Group etc.) from which a creature's behavior, sensory and motor system is constructed. While we take an object-oriented approach, our philosophy has been to attempt to reduce the amount of subclassing necessary to create a new creature. This is done in primarily two ways. First, we rely on a callback mechanism rather than subclassing to change the default behavior associated with certain classes. That is, when an object is likely to need to do something which is specific to a creature or instance of the object (e.g. the find or filter phase of a Releasing Mechanism), we implement the functionality as a C-callback which is called by the object. Thus, if different behavior is desired, one simply installs a new callback. This typically involves far less overhead than defining a new class. Second, we make extensive use of parameters to specify creature or instance-specific values for fields.Thus, we err on the side of providing parameters if by doing so we can avoid having to subclass.

The Hamsterdam tool kit represents about 30,000 lines of C++ code, and approximately 150 classes. By contrast the amount of code required to create a new creature is rather small. Silas, for example, is implemented in rough 5000 lines of code, with close to 3000 used to implement his dog specific motor skills. Dr. J. Puppet represents about 1800 lines of code. The long suffering hamster by contrast is only 300 lines of code. The interested reader is urged to consult the Appendix for a summary of the major classes which the Hamsterdam tool kit defines.

5.9 Summary

In this chapter we have presented our computational model for interactive creatures. We presented a layered architecture consisting of geometry, a motor system to manipulate the geometry over time and a behavior system to tell the motor system what to do. The remainder of the chapter was organized around the five key problems presented in chapter 3. Below, for each problem to be addressed we summarize the relevant aspects of our computational model.

Relevance: Do the right things...

- The Behavior System is a distributed system of goal-directed, self-interested entities called Behaviors.
- Behaviors are responsible for evaluating their own relevance based on input from Internal Variables (i.e. motivations) and Releasing Mechanisms (i.e. mechanisms which transduce a value from the presence of specific stimuli in the environment).
- Internal Variables and Releasing Mechanisms use the same currency to represent their value, and this allows the Behavior to combine their values in meaningful ways.

Coherence: Show the right amount of persistence...

- Behaviors are organized into groups called Behavior Groups. Behavior Groups in turn can be organized in loose hierarchies. Within a given Behavior Group, the behaviors often represent alternative strategies for accomplishing a particular goal. In any event, Behaviors with a Behavior Group compete for control.

- They compete using the Ludlow/Minsky model of mutual inhibition. Via the “avalanche effect” this model insures robust arbitration. Inhibitory Gains are used to make a Behavior more or less persistent, and Level Of Interest is used to prevent inappropriate persistence. Thus, it provides “knobs” to help control persistence.

Convey Motivational State: Its not what you do, but how you do it...

- The Motor System is designed to support coherent concurrent motion. Constructs called Degrees Of Freedom (DOFs) control access by Motor Skills (e.g. “walking”) to the underlying geometry (e.g. joints). They insure that only one Motor Skill may have access to a given DOF at a time. A Motor Skill can only become active if all of its DOFs are available.
- The MotorController supports three imperative forms for motor commands. This allows Behaviors in the Behavior System to issue commands with varying levels of importance. Winning Behaviors in the competition for control issue primary commands (“do it if possible”), whereas losing Behaviors issue secondary and meta commands which essentially represent suggestions (“do it if no one objects”, or “if you do it, do it this way”).
- This design of the Motor System allows multiple behaviors to express their preferences for motor actions while still insuring coherent, concurrent motion. This approach facilitates the display of motivational state.

Adaptation: Learn the right things...

- Our approach to learning focuses on two types of learning: learning how to re-use old behaviors in new contexts to satisfy previously un-associated motivational variables (i.e new instrumental contigencies), and learning how to better predict known contexts (i.e. new classical contingencies).
- Learning is driven by motivational variables. When they undergo a significant change they seek to explain the change in terms of things which have recently changed in the environment, or in terms of their own actions.
- By looking at short-term memory, the system builds Releasing Mechanisms called BSRMs which fire when they detect a conjunction of behavior and stimulus (B && S). These Releasing Mechanisms compete within Discovery Groups to explain the change in the motivational variable using the Barto-Sutton Temporal Difference equation. This equation is used to learn the appropriate MaxValues for the BSRMs in the Discovery Group.
- In addition to learning the appropriate value for the BSRMs, the system keeps track of their reliability. When the value and reliability gets above threshold levels, the BSRM is expanded into a Behavior and its associated Releasing Mechanism and this is added to the motivational variable’s set of appetitive behaviors. This mechanism closely follows Lorenz’s conceptual model for animal learning [Lorenz73].
- To separate classical contingencies from instrumental contingencies it is necessary to augment the system with BSRMs which fire on ((!B) && S). Thus, for every conjunction, the system builds two BSRMs: (B && S) and ((!B) && S). If it is true that

for a given S within a Discovery Group that (a) for each BSRM pair the reliability of ((!B) && S) is higher than that for (B && S), and (b) the value of at least one ((!B) && S) is above a threshold, then the system assumes that S has incremental predictive value, but it comes from predicting the occurrence of a stimulus which reliably follows S. Thus, the system finds the closest behavior which has a known instrumental contingency and adds to its set of Releasing Mechanisms a new Releasing Mechanism which fires on S. This helps the system learn classical contigencies.

External Control: Pay attention to the guy who pays your bills...

- The design of the system makes it easy to integrate external control at multiple levels of abstraction.

We concluded the chapter with a discussion of our approach toward sensing. We saw that there are several forms of sensing available to creatures in our system and we argued that synthetic vision is a particularly promising approach for facilitating low level navigation and obstacle avoidance. We presented a technique, derived from ideas from ethology, that recovers a gross measure of motion energy from a scene and uses it to control movement.

The computational model presented here has been used to build a number of interactive animated characters, including Silas T. Dog. Our experience in building these characters and using them in interactive installations such as ALIVE over the past three years, together with a discussion of specific experiments which highlight key aspects of our model is the topic of the next chapter.

6.0 Results

The computational model presented in the previous chapter has been tested in essentially two ways. First, we have built a number of creatures, including Silas T. Dog and have used them in the context of the ALIVE project [Maes96] and Tinsley Galyean's DogMatic project [Galyean95Galyean95]. The experience of building these characters and subsequently watching them interact with several thousand users over the past three years has provided us with useful insights into the strengths and weaknesses of the approach. Fundamentally though, our experience has convinced us that the system is capable of generating robust and believable behavior in a dynamic and unpredictable environment. Second, we have performed systematic tests of key aspects of the algorithm in order to better understand their effect on the resulting behavior.

We will begin this chapter by describing the ALIVE system and the creatures we have built for it. The goal here is to give the reader a sense of the range of characters and behaviors which have been addressed using our model, and to highlight instances where specific aspects of the approach proved useful. We will then briefly describe the results from a series of tests which demonstrate certain aspects of the action-selection and learning mechanisms. We will conclude the chapter with a discussion of the limitations of the approach and with our perspective on several important, but open, questions.

6.1 Creatures In ALIVE

The architecture described in chapter five has served as the basis for all of the creatures implemented as part of the ALIVE project [Maes96]. ALIVE ("Artificial Life Interactive Video Environment") is a system which allows wireless full body interaction between a participant and autonomous graphical creatures in a 3D world. The fundamental metaphor of the ALIVE system is that of a "magic mirror" in which the user sees herself and the graphical creatures which co-exist with her in the world of the magic mirror. Computer vision techniques [Wren95] are used to separate the user's image from the background, and to recover the 3d location of the user's head, hands, and feet. In addition, the vision system does basic gesture and posture recognition. The creatures in the ALIVE world respond to the participant, in real time, on the basis of this information and their own internal motivations. Figure 36 gives two examples of Silas responding to a user's gestures.

Several thousand people have experienced the ALIVE system. It was an invited installation at Siggraph '93 and '95, as well as one of three installations at the AAAI Art Show in 1994. It was installed in the City Art Museum of Nagoya Japan for 3 months as part of ARTEC '95. It has been one of the standard Media Lab demos since 1994. Users have ranged from the Vice President of the United States, to a 90 year-old grandfather in Japan to a three year old child in Los Angeles. The system has proved to be robust and an enjoyable and novel experience for the participants.

There have been several worlds in the ALIVE system. In the first ALIVE installation a user could move between two worlds, a puppet world in which the principal character was Dr. J.T. Puppet (see Figure 37) and a hamster world in which there were two characters, a hamster and a predator. In the more recent versions the user interacts with either the puppet or Silas T. Dog. In the following sections we will describe these characters in more detail.

Figure 36 Silas in ALIVE

Here are two examples of Silas interacting with a user in the ALIVE space. On the left he is sitting in response to the user pointing down. On the right he is begging, this time in response to the user's hands over their head.

Figure 37 Dr. J.T. Puppet

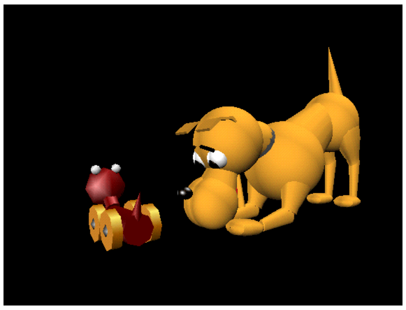
Dr. J. T. Puppet is usedboth in the ALIVE system and the STIVE system . He is also used in as the teacher for Silas in the learning experiments. The original model for Dr. J.T. Puppet was done by Johnny Yoon.

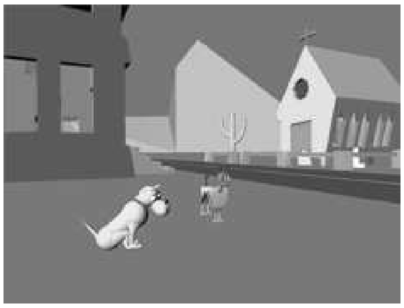
6.1.1 Dr. J.T. Puppet

Dr. J.T. Puppet has had several incarnations[27] but in the first system he was largely a reactive character (i.e. he had no internal motivations to speak of). He would follow the user around in the space, and imitate some of the user's gestures. If the user pointed left or right, he would go off and sit in a corner. If the user waved, he would jump up and return. He had a set of facial expressions which were used to convey his emotional state. He would look sad if he was sent away, relieved if was told to return, and would grin if he was tickled. We have recently re-implemented the puppet, and consistently find that

27. Lenny Foner, Jeremy Brown, Johnny Yoon, and Cecile Pham were all largely responsible for the early incarnations of the puppet. Liz Manicatide helped with the facial animation, and did the original graphical concept for Silas T. Dog.

Figure 38 Silas and some of his friends





.On the left, we have his long-suffering friend the Hamster. On the right he is on the set of DogMatic with his half-brother "Lucky", the star of DogMatic.

although his behavioral repertoire is very limited people find him very engaging (in no small measure due to excellent modeling by Johnny Yoon).

The latest version of the puppet has been used in a variety of environments in addition to ALIVE. He has been used in the STIVE project [Azarbayejani96] to mimic the user's gestures. Here he uses the 3d locations of the user's hands and head, as recovered by the stereo vision system, as targets toward which to move his hands and head using his built-in inverse kinematics. We also use him as Silas's teacher (we awarded him an honorary degree when Silas objected to being taught by a common puppet).

In each instance, the puppet has a behavior system which controls his actions. In the case of STIVE it is very simple: it uses Releasing Mechanisms to detect the position of the respect body parts, and he has a set of behaviors which move his limbs appropriately. In his role as teacher, he has a number of parameters including the behavior to train and the gesture to use. His behavior system is set up so that when a Releasing Mechanism detects that Silas is beginning to perform the desired behavior (in this training paradigm, the puppet waits until performance of the desired behavior), the puppet responds (i.e. his "doTraining" behavior becomes active) by performing the appropriate gesture, and a few ticks later by rewarding the dog. He has another behavior which tests whether Silas has indeed learned the trick, and another which he uses for teaching a behavior as part of a chain.

6.1.2 Harold T. Hamster

The hamster and predators of the hamster world were more complex. The hamster engaged in a number of activities, one of which was foraging for food. Food, however, was only available if the user went over to a virtual table and got it for him. Thus, the hamster would follow the user around in the space and beg for food. If the user bent over, the hamster would roll-over to have his tummy scratched. However, if this persisted too long, he would get bored and do something else. There was also a predator in the world who could be let into the hamster's enclosure by the user pressing a virtual button. The predator would attempt to chase and eat the hamster, and the hamster would typically attempt to flee. However, to complicate things, the predator viewed the user as his predator and would attempt to avoid the user. Similarly, the hamster's desire to flee competed with his desire to get food or be patted by the user. The user could choose to help the hamster and herd the predator back into its cage, or stand back and let the predator catch the hamster.

The most engaging aspect of this world, apart from the life and death drama of the scenario, was the interplay of the competing desires of the hamster and the predator. The weakness of the demo was that the hamster and predator did not convey their motivational state, other than in their pattern of locomotion (e.g. moving rapidly away from the predator was interpreted as fleeing from it). In addition, it was not intuitively obvious how one should interact with a hamster. For these reasons we moved to Silas T. Dog.

6.1.3 Silas T. Dog

In the latest version of ALIVE the user interacts with Silas T. Dog. Silas responds to the user based on the user's gestures as well as his own motivational state. He has his own agenda which includes drinking, eating, peeing, playing with the user, chasing a hamster, and sleeping. Silas responds to a dozen or so gestures and postures of the user, and responds appropriately (e.g. if the user bends over and holds out her hand, Silas moves toward the outstretched hand and eventually sits and shakes his paw). The dog always looks at its current object of interest (head, hand, etc.), and his tail, ears and head move appropriately based on his level of happiness. Silas has a number of internal motivations which he is constantly trying to satisfy. For example, if his desire to fetch is high, and the user has not played ball with him, he will find a ball and drop it at the person's feet. Similarly, he will periodically take a break to get a drink of water, or satisfy other biological functions.

Silas conveys his motivational state and intentionality through his gait, gaze, posture and position of his ears, head and tail. For example, Silas has an Internal Variable which represents his level of happiness. A number of subsidiary behaviors use the level of this variable to determine what motor commands they issue (note, the magnitude of the variable is such that it is all but guaranteed that they will be winning behaviors, and in any event they always issue secondary or meta-commands). Thus, the ear-position behavior will move the ears back if the level of the variable is low, and forward if it is high. Similarly, a behavior responsible for setting the gait, chooses a gait based on the level of this variable. Silas also uses Internal Variables to represent parameters such as his level of aggression or level of guilt. Typically, events in the world set the state of these variables and they decay over time to their normal level. For example, when the user points left or right, the Releasing Mechanism which detects this gesture sets the level of happiness to zero (i.e. very-sad). Or in another case, if Silas "knows" that the user is performing a gesture, but he is choosing to ignore it, the level of his guilt variable is set high and he attempts to look guilty. Lastly, a high level of aggression is caused by the presence of the hamster and stays high until the dog catches the hamster and shakes him. However, if the user sends Silas away, perhaps indicating her displeasure at this wanton display of violence, his level of aggression drops.[28]

Silas always indicates his current object of interest by gazing at it. His eyes always move first, followed by his head and neck and finally his body.

He utilizes both direct sensing and synthetic vision. Synthetic vision via a synthetic vision sensor is used for low level navigation and obstacle avoidance. He also has a "sniff" sensor on his nose which is responsible for most of his direct sensing. In addition, he has sensors on his mouth and hips to facilitate certain behaviors (e.g. his "hip" sensor tells him when he is close enough to the fire hydrant to lift his leg). These subsidiary sensors are simply for convenience since by using them we can avoid performing explicit transformations from "nose" to "hip" coordinates, for example. See 5.7 for details.

Silas is one of the most sophisticated autonomous animated creatures created to date. Figure 39 provides some summary statistics for the version of Silas which is used in the

Figure 39 Summary Statistics for Silas

Silas Behavior System:

- 90 Unique Behavior Nodes (60 Leaf Nodes)
- 23 Unique Behavior Groups
- 86 Unique Releasing Mechanisms
- 28 Unique Motivational Variables

ALIVE system and see the Appendix for a detailed list of the Silas's Motor Skills and Degrees of Freedom as well as the Motor Commands to which he responds. Silas runs at 15-20Hz on a Onyx Reality Engine with rendering and sensing time (i.e. the second render for his synthetic vision) comprising most of the update time. The evaluation of the Behavior System itself typically takes less than 5 milliseconds.

6.1.4 The "Invisible" Person and Creatures of a Lesser Kind

One of the surprising lessons of ALIVE was that the architecture which we developed was so flexible and easy to use (at least for its creator) that almost all of the objects in ALIVE incorporate a behavior system of greater or lesser complexity. Here we list some of the other creatures:

- **The Invisible Person**. For example, in order to implement an interface to the vision system in a way that would be transparent to all of the creatures in ALIVE, we created a creature which all of the creatures could see, but which is invisible to the user. This creature is largely driven by the user's actions. On each time step, it queries the vision system for the locations of the hands, feet, head and centroid of the user's body as well as for the vision system's interpretation of the user's posture and gesture (e.g. "sitting" and "hand-extended"). It then sets the locations of its respective body parts as appropriate, and updates its creature-accessible fields (i.e. the fields used for sensing by Silas and other creatures) to reflect the user's posture and ges-

28. During a demo with Alan Alda, Silas grabbed the hamster and was giving him a hard shake when Alan spoke sternly to him and pointed for him to go away. Silas dropped the hamster and moved off, looking appropriately contrite. However, as he passed the hamster he turned and growled (presumably because his level of aggression went up because he was so close to the hamster) then continued on his way. For Alan, this was the most impressive part of the demo because it demonstrated a very life-like interaction of emotions.

ture. However, it also has a behavior system which updates itself after acquiring the vision data. The behavior system is responsible for picking up the ball if the user's hand is close to the ball, carrying the ball, and releasing the ball if the user performs a throwing gesture. While this is very simple behavior, we found it very convenient to be able to be able to specify it via a text file, and the structure of Releasing Mechanisms and Behaviors are ideally suited to this kind of problem: time-varying reactive behavior in a noisy environment.

- **A Ball.** The "Ball" in ALIVE also has a behavior system whose purpose is largely to simulate simple physics: when it is on the floor it does nothing, when it is in the air it drops unless it is being held by another creature.
- **An Experimental Apparatus.** The "experimental apparatus" used to train the hamsters in some of the learning experiments described later is also a creature with a behavior system. In this case the behavior system is responsible for deciding when to start performing the experiment (e.g. when a motivational variable reaches a certain level), for performing the experiment, and for deciding when to end the experiment (e.g. it turns off the lights and the shock) when a Releasing Mechanism signals that the Hamster has moved off of a simulated electrified mat.
- **A Bird.** In our Siggraph '95 installation we used an autonomous bird as the mechanism by which a user could switch from one world into the next. The behavior system of the bird was such that it would discretely circle around the user's body and head. However, if the user's hand was above their head, the bird would fly near it and change its color over the course of a few ticks. If the user's hand remained above their head, the bird would cause the worlds to change.[29]

The behavior in these examples is all very simple and could have been done in any number of ways. The point we wish to make here is simply that using our approach: "simple things are simple, and hard things are possible".

6.1.5 Distributed ALIVE

ALIVE and the creature architecture described in this thesis was originally intended to be a stand-alone system. Last year, however, we implemented a distributed version of the system in which users in two different locations could interact with each other and one or more autonomous creatures [Russell96]. One of the design criteria was that it work well in a low-bandwidth, high latency environment, such as a wide-area network. This placed a premium on an efficient protocol for updating the state of a creature as well as on a model which distributed the computation across the different locations (i.e. we wanted to avoid one central "behavior server").

The relevant lesson from this work was that our architecture made the implementation rather straightforward. This was due in large measure to the role of the Motor Control-

29. There were four worlds: a normal world, a world in which the user became a large animated dog bone, another in which they became an animated fire hydrant, and a fourth in which they became a video avatar (the latter was the brain child of Trevor Darrell). Silas responded to each of these instantiations of the user differently: he would play with the normal user and the astronaut, chase and eat the dog bone, and pee on the hydrant. This was very easy to implement as we simply changed the "creatureType" field of the object which represented the user. To insure the appropriate response by Silas, we adjusted the level of the appropriate motivational variables accordingly

ler, as a bottleneck between the Behavior System and the Motor System. See [Russell96] for the details but the basic idea was that for any given creature, there was one master and a set of drones (one per remote site). However, only the master's behavior system was active. Thus, the master alone was responsible for sensing and deciding what to do. However, whenever the master issued a motor command, its Motor Controller would broadcast the command to the drones who would pass it on to their respective Motor Controllers for execution.This was a much more efficient means of communicating than the alternative of passing transforms and coordinates. To put this point in perspective there are 135 transforms in Silas. If on any given tick half of them change as a result of motor actions, this represents approximately 9000 bytes (70, 4x4 matrices of 8 byte doubles) which would have to be sent over the wire. By contrast the data structure used to communicate motor commands is less than 100 bytes. Even if ten motor commands are issued on a given tick, this is still an order of magnitude more efficient than passing transforms. It was also very easy to implement because, to a first approximation, it only required changes to the dispatch mechanism of the Motor Controller. It is, in fact, transparent to the Behavior System whether it is running in a stand-alone or distributed environment. Similarly, from the standpoint of another creature or user, there is no difference between a master and a drone.

Distributed ALIVE was demonstrated at Siggraph '95, and there are permanent installations at the Media Lab, British Telecom in the UK, and ATR in Japan.

6.1.6 Summary of our experience building creatures for ALIVE

Our experience over the past three years with the ALIVE project has demonstrated that the framework presented in this thesis is capable of generating believable and robust behavior in a highly dynamic and uncertain environment. As one person put it: "Silas has a presence". Indeed, people generally find Silas both engaging and perhaps even "lifelike". His behavior seems explicable to most people and in giving hundreds of demos, we have been rarely asked: "so why did he do that?". We have generally succeeded in making his internal state visible. People know when he is happy or sad, and why.

Our experience in constructing Silas and other creatures for this environment validates many of the aspects of the architecture. The arbitration mechanism has proven fast and robust. Silas, mostly, does the right things for the right amount of time. He rarely, if ever, demonstrates dithering or pathological persistence. Varying Level Of Interest and unique Inhibitory Gains, while used sparingly (most of the time the default values suffice) are very useful in particular situations as described earlier. The ability of the motor system to support coherent concurrent motion, the use of multiple imperative forms for commands, and the mechanism whereby subsidiary behaviors may express their preferences for motor actions has proved essential for conveying motivational state. The use of Behavior Groups has made it straight forward to add new behaviors. Our approach to sensing and use of synthetic vision for low-level navigation and obstacle avoidance has proven fast, simple, and robust. The fact that we have used the system to construct a wide variety of creatures suggests that (a) the architecture is extensible and (b) it is easy to build new creatures. This is attributable in part to the object-oriented architecture, in part to our approach to extensibility via extensive use of callbacks and parameters, and in part to the ability to share behaviors between creatures.

6.2 DogMatic

We have also developed a number of creatures which are used in the context of DogMatic, an interactive story system developed by Tinsley Galyean of the Media Lab [Galyean95]. Galyean's system features a computational director which provides "direction" to the creatures so as to meet the requirements of the story. For example, at the beginning of the story, a dog hops out of a car and wanders around. If the user, who is wearing a head-mounted display, does not pay attention to the dog, the director will send the dog over to the user. If the user still does not pay attention, the director effectively tells the dog: "the user's leg is a fine replacement for a hydrant, and you really have to...". The resulting behavior on the part of the dog usually captures the user's attention.

There are several creatures in Dogmatic, including a dog named "Lucky" and a car. While both the car and Lucky were autonomous, Galyean used the all of the techniques described in section 5.6 to direct Lucky's behavior. To encourage Lucky to "pee", his computational director would increase Lucky's NEEDTOPEE motivational variable. To direct Lucky to use the user's leg instead of a tree, the director altered the Releasing Mechanism associated with "peeing" so that it would fire on the presence of a leg as opposed to a tree. This incidentally was facilitated by the use of "pronomes", because the approach behavior in this case used the pronome set by this Releasing Mechanism. In other cases, the director would directly issue motor commands.

6.3 Experiments

Here we wish to review several experiments which highlight particular aspects of the architecture.

6.3.1 Action-Selection

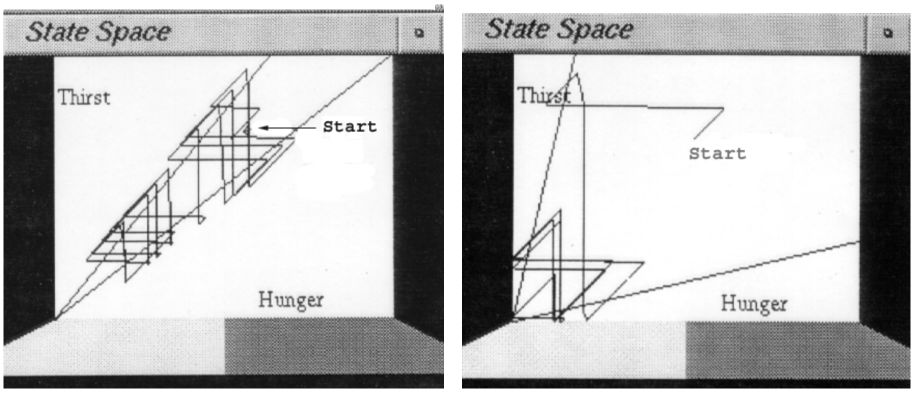
This section uses results from Hamsterdam and Alive to demonstrate the importance of some of the ideas incorporated in the computational model.

In the cases presented below, the Hamster is in an enclosure containing food and water. To eat or drink, it must explore the enclosure until it senses the food or water, move to the appropriate resource, position its head accordingly, and then chew or drink. Thus, the various activities have the equivalent of appetitive and consummatory phases. Figure 40 demonstrates how inhibitory gains may be used to control persistence. The figure presents state-space diagrams [after McFarland] for hunger and thirst levels under 2 different cases of inhibitory gains (low and high). The straight diagonal lines represent switching lines based on the level of inhibitory gains. The starting point for the systems is marked, and the origin represents the point of satiation for both thirst and hunger. As one can see, when the gains are low, the system tends to dither between feeding and drinking with the result that the system takes longer to reach satiation levels. At higher gains, the active activity shows more persistence with the effect that satiation is reached sooner.

Figure 41 demonstrates the use of Level Of Interest to provide a form of time-sharing by showing the pattern of behaviors over time when activity-specific Level Of Interest is included. The various blocks correspond to when a given activity is active. Feeding is represented by the black blocks, drinking by the dark gray blocks and cleaning by the lighter blocks on top. The initial level of hunger is twice that of thirst, and the need to

Figure 40

Inhibitory Gains Help Control Persistence



This figure demonstrates the effect of varying levels of Inhibitory Gains on persistence. On the left, the gains are close to 1.5, and as a result the creature dithers between eating and drinking. On the right, the gains are much higher and as a result the creature persists at a given activity longer. The effect is that it reaches satiation of both motivations more quickly.

clean is half of the level of thirst, and there is neither food nor water to be had. Without Level Of Interest drinking and cleaning would never become active. However, as a result of Level Of Interest the system alternates between looking for food and looking for water with an occasional interruption for cleaning, even though the internal value (before Level Of Interest) of the various activities stay unchanged. This is an important demonstration because it shows how the system can avoid the mindless pursuit of an unattainable goal to the detriment of other goals.

Figure 41

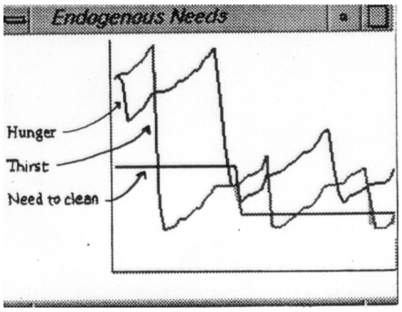
Level Of Interest Promotes Time-Sharing

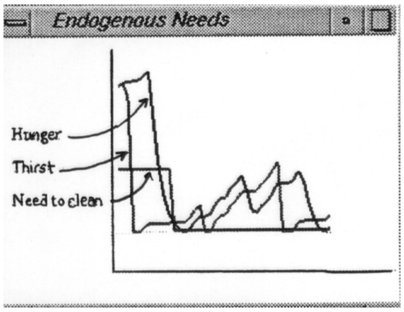
This figure demonstrates how Level Of Interest can produce a form of time-sharing. Here we see a trace over time showing the active behavior. Feeding (i.e. looking for food) corresponds to the lowest set of blocks, drinking (i.e. looking for water) the middle set, and cleaning the upper set of blocks. In the experiment, the level of the motivational variables remain unchanged because there is no water or food in the enclosure and cleaning has no effect. Since hunger is has the highest value, without level of Interest you would not see an alternation of behaviors. With Level Of Interest, however, even if the intrinsic value of the behavior remains unchanged, its value after Level Of Interest goes down as the behavior is active, thus eventually causing a switch to another behavior.

Figure 42 demonstrates how adjustments to the range associated with a given Releasing Mechanism can result in opportunistic behavior. The figure show the levels of hunger and thirst over time. In both cases, the Hamster starts at the same location and in the process of searching for food passes near water. In the graph on the left, with a lower allowed maximum for the water releasing mechanism, the Hamster ignores the water until after it has eaten. When a higher value is used (the graph on the right), the Hamster

Figure 42

Opportunistic Behavior





These two graphs demonstrate how varying the allowed maximum for a given Releasing Mechanism can affect the level of opportunistic behavior. See text for details

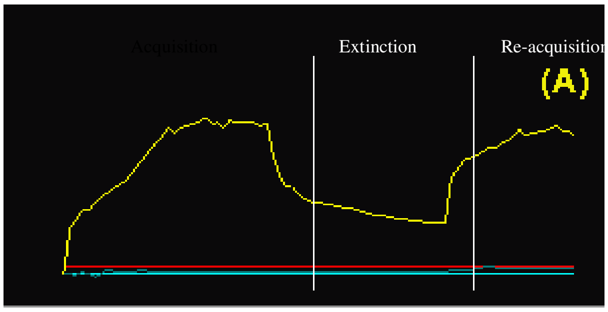
interrupts its search for food to take advantage of the water. This can be seen by comparing the respective traces of endogenous needs and noting that the level of thirst drops sooner in the graph on the right, than on the left.

6.3.2 Learning

Here we demonstrate some of the key features of the learning algorithm described in the computational model In the examples that follow (experiments with the dog), the training protocol is that the dog approaches the puppet and begins to cycle through several of his behaviors (SIT, BEG, CROUCH, LIE-DOWN, etc.). The puppet acts as a trainer and when it detects that the dog is beginning to perform a desired behavior, the puppet performs the training gesture (e.g. “handExtended”) and after a few ticks gives the dog a bone if the dog is still engaged in the desired behavior. For the purposes of these experiments, Silas has a simplified behavior system. That is, he has a Feeding sub-system which contains an appetitive “find food bowl and approach” behavior and a consummatory “Chew” behavior. Silas also has an exploratory Behavior Group that causes him to

Figure 43

Trace of BSRM for “HandExtended && Sitting” during training



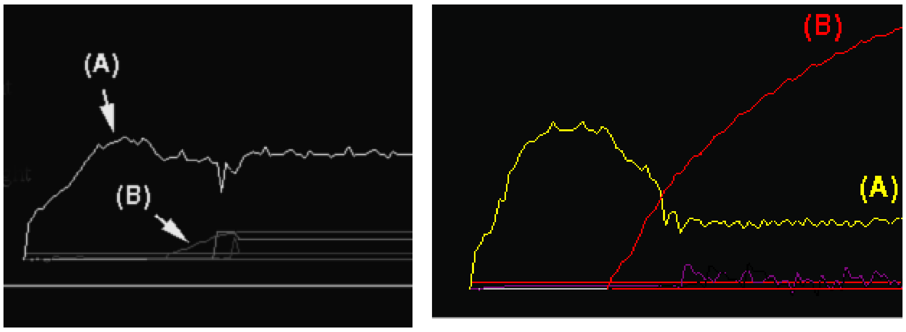
Here we see a trace of the value of the “HandExtended-SIT” BSRM during acquisition, extinction and re-acquisition.

approach the user and to engage in a variety of his different behaviors. He has a motivational variable “NEEDTOEAT” which drives the learning process.

In Figure 43 we show the trace for the “HandExtended && Sitting” BSRM during acquisition, extinction, and reacquisition. Extinction begins after 50 trials, so one can see that after about 20 trials the value for the BSRM has stabilized. One will note that the initial change in value during extinction, and re-acquisition is rather high due in part to a high learning rate (remember that the learning rate is based on the reliability contrast so when reinforcement is reliably reliable or reliably un-reliable the learning rate will be high).

In Figure 44 we show two examples of “blocking”. On the left, we show the traces for an experiment in which the trainer initially starts training one gesture-behavior pair (“HandExtended-SIT”).This is shown by the yellow line (A) in the figure. After the

Figure 44 Demonstration of Blocking



The learned value associated with a stimulus behavior pair is a function of its “incremental predictive value. On the left, (B) is blocked by (A) because it has no incremental predictive value (see text for details). On the right, by contrast (B) does have predictive value and this is reflected in its dramatic increase in value at the expense of (A).

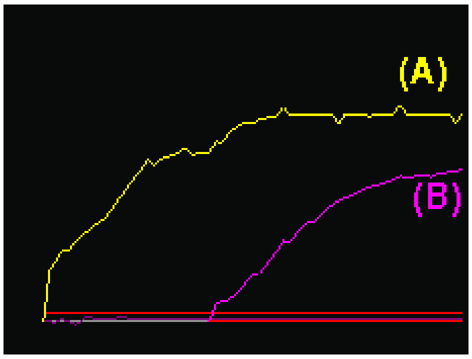
“HandExtended-SIT” approaches asymptote, the puppet begins to “Jump” at precisely the same time that he extends his hand. This is shown by the red line (B). The key thing to note is that the value of “Jump-SIT” BSRM never gets very high. This is a consequence of being “blocked” by “HandExtended-SIT”, which already adequately predicts the future reinforcement. As a result, Silas attributes little value to the “jumping” component of the combined gesture, because it has no incremental predictive value beyond the “hand extended” component. This is a typical training phenomena found in animals.

On the right, we repeat the experiment except this time the puppet begins to jump one tick before he extends his hand, and this has a dramatic effect on the resulting behavior. Since “Jump-SIT” now has incremental predictive value (it predicts future reinforcement as well as “HandExtended-SIT”), it rapidly increases in value at the expense of “HandExtended-SIT” which correspondingly loses value. This once again is an effect that one sees in nature.

The learning dynamics demonstrated in Figures 43-45 are a direct consequence of using the Barto-Sutton learning model, and thus are not particularly surprising.

Figure 45

Demonstration of Chaining

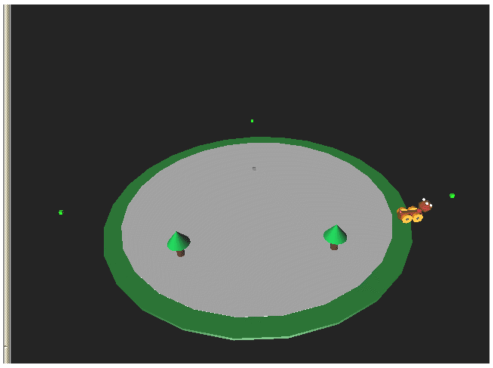


Here we see a demonstration of chaining. Trace (A) corresponds to “HandExtended-SIT” BSRM, and trace (B) to “Hand-Pointing-BEG” BSRM. (A) is trained first with the bone being given as reinforcement. Once this trick is learned, the trainer starts training (B), except here the reinforcement for (B) is “HandExtended” and not the bone.

Figure 45 demonstrates chaining via operant conditioning. In this demonstration, the dog is first trained to sit when the puppet’s hand is extended, with a dog bone being the reinforcement. This rapidly gains value as shown by curve (A). Once it approaches asymptote, the puppet works on the behavior which is to precede sitting. In this case the dog is rewarded for Begging in response to a pointing gesture by the puppet. However, in this case the reward is not food, but the appearance of the “hand extended” gesture. Thus, “hand extended” has become a conditioned reinforcer for “handPointing-Begging”. This is shown by curve (B).

Figure 46

Hamster learns to anticipate shock and flee before receiving shock

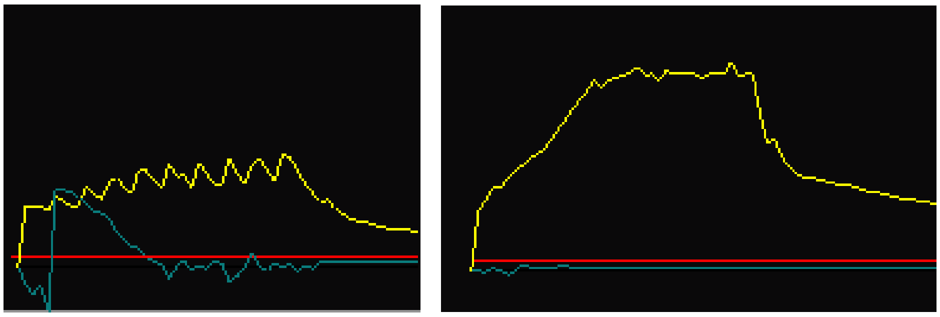


Here the hamster has learned that the green light predicts a future shock and so it begins to flee when the green light comes on. See text for details

In the experiment shown in Figure 46 the hamster learns to both anticipate an event (i.e. a shock), and to respond in the appropriate manner. This requires him to learn both an operant and classical contigency. The Hamster is engaged in foraging in an open pen. The floor of the pen (i.e. the gray disk) can be “electrified” and its color can change. The experiment is set up so that the floor color will go from grey to green to yellow to red. When the color is red, a shock is delivered for up to 40 ticks as long as the Hamster remains within a certain radius of the center of the pen. If the Hamster leaves the critical area, the shock is immediately turned off. The Hamster has a built-in response to shock: it will randomly choose to either “freeze” or to “run away.” It also has a built-in association that the shock comes from the environment (i.e. when it is shocked the environment becomes its object of interest). The motivational variable is “relief from pain.” Since the Hamster's speed is such that it can leave the pen in substantially less time than the shock's duration, running away should be the preferred action, and it should have an increasing tendency to run away as the floor-color progresses from green to red.

Over the course of 20 trials (i.e. 20 individual shocks), the Hamster learns that running away brings a quicker reduction in pain than does freezing, and so running away becomes the preferred option. However, the most interesting aspect of the experiment is that he also learns to anticipate the shock and run away when the green light comes on. This is a direct consequence of learning that there is no operant contingency associated with the sequence of green and yellow lights but there is a classical contingency between the green and yellow lights and the shock. Typically he is engaged in a behavior associated with foraging when the lights come on, so the associated BSRMs (e.g. ((forage) && (green light))) will take on a positive value. In fact, if the system did not compare the relative reliabilities of (S && B) and (S && (!B)) and just based the decision on the value of the BSRM going over a threshold value, the Hamster would learn superstitious behavior (e.g. find food when green). By virtue of the comparison of reliabilities which indicates that the reliability of (S && B) is no higher than the reliability of (S && (!B)) the Hamster discovers that the association is coincidental. But since the value of the BSRM is high, the stimulus must have incremental predictive value. Thus it learns the association between the time course of the colors and being shocked and begins to flee when it senses the green light and successfully learns to avoid the shock.

Figure 47 Reinforcement Ratio’s Effect on Learning Rate



On the left the reinforcement ratio is 1 to 3 (rewarded every third time) whereas on the right the reinforcement ratio is 1 to 1

In Figure 47 we show the traces over time of the strength of a BSRM under two training regimes. On the left, the Dog is rewarded every third time, whereas on the right he is rewarded every time he responds appropriately to the gesture. In both cases, after the Dog has learned the trick, it is extinguished (i.e. he is not rewarded). The key point to note between the two graphs is that the slopes of the traces are different, particularly during the initial phases of extinction. This reflects the effect of the reliability-based learning rate. Since the reliability of the behavior in producing the reward is lower when it is rewarded every third time, the learning rate is lower as well.

6.4 Open Questions

As with any approach, there are a number of limitations which must be considered when evaluating its usefulness for a given application. In the previous section we touched on issues which weren't currently addressed by the architecture, here we consider a number of open questions.

- **Is it too complicated?** The model presented is rich and flexible, but there is no shortage of concepts (e.g. Releasing Mechanisms, Behaviors, Behavior Groups), and parameters (e.g. Inhibitory Gains, Level Of Interest, maxValues associated with Individual Releasing Mechanisms etc.) Are they all necessary, and in the case of the parameters do they all need to be set? We have argued that it is important to have some sort of explicit mechanism for providing control over persistence and relevance. However, it is also important to note is that in most cases the default values for the parameters work fine. For example, in Silas, only 17 out of 90 behavior nodes specify Inhibitory Gains other than 2.0, and only 3 behaviors utilize a variable Level Of Interest. Level of Interest is a good example of a concept which need only be used sparingly, but is extremely powerful when you need it. In addition, one of the goals of learning is to learn the appropriate maxValues for Releasing Mechanisms and Level Of Interest for behaviors. Lastly, because the interactions of the behaviors is localized, once a Behavior Group has been tuned, it is usually a simple matter to use it in other contexts. Our use of multiple imperative forms for motor commands has proved essential in the design of characters such as Silas, and it is our opinion that it is a simpler approach than the "command fusion" approaches of Rosenblatt [Rosenblatt89] and Tyrrell [Tyrrell93].

  This said, it is important to stress that "real people" like animators will want a layer above and below the architecture described in this thesis. Below the architecture they will want interactive tools which allow them to fine-tune the quality of the motion. Above the architecture they will want to work at the "find the hamster and pick it up and put it back in its cage" level and let the system generate the underlying behavior network necessary to produce that kind of behavior.

- **What about planning?** The system does not incorporate planning or reasoning. Two questions follow from this omission. First, what are the limitations on the resulting behavior that result from not incorporating these features? Second, how easy would it be to incorporate one of more of these features? With respect to the first question, it should be recognized that this issue of planning and reasoning vs. pre-organized behavior augmented by learning and memory is at the heart of the debate between cognitive and classical ethologists [Bekoff96, Griffin94, Gould94]. For example, a classical ethologist might argue that even the apparently sophisticated behavior of beavers can be largely be explained as pre-organized behavior augmented by learning, whereas a cognitive ethologist such as Griffin would argue that the most parsimonious account would include reasoning on the part of the beavers. Clearly, we are not going to resolve this debate here in a few short paragraphs other than to make two points. The first is that the appearance of “planned behavior” does not necessarily imply an ability to plan. For example, when Silas “wants to play ball”, it finds and approaches the ball, picks it up and carries it back to the user and drops it at their feet. An observer might attribute planning to Silas, when in fact this behavior is an artifact of how his behavior system is organized. Similarly, with the addition of a memory model which allowed him to remember where he had been, one probably could write a sub-system which would allow him to play “hide-and-seek” or “visit-N-locations and avoid the predator” (note, the lack of such a memory model is a significant limitation which needs to be addressed sooner rather than later). Animal navigation is another area where it isn’t clear whether the seemingly remarkable navigational ability of animals is due to the acquisition and manipulation of cognitive maps or more due to clever hacks (e.g. path integration) augmented by use of landmarks to a greater or lesser extent. The second point, though, is that as these characters are called on to perform increasingly human-like tasks (e.g. “shop and cook dinner for the kids”, or “do these 5 errands before 4:00PM”), the easiest or most efficient approach may be to integrate a planner.

It is perhaps more fruitful to address the second question, namely how easy would it be to integrate a planner into this system, should one choose to do so? The short answer to this question is that it should be straightforward for the very same reasons that it is straightforward to integrate external direction. While the tacit assumption in our discussion of external control was that the external entity was a director (human or computational), it could just as easily be a planner. Indeed, the architecture provides a rich interface that would allow a planner to be integrated at a number of different levels. The system’s use of pronomes would also seem to facilitate the integration of a planner by allowing pre-organized or even compiled-on-the-fly collections of behaviors to be applied to an object of interest whose value is bound at run-time. One can also imagine certain behaviors being written so as to use the plans generated by a planner as “resources for action” as proposed by Payton [Payton90] or Chapman [Chapman90]. For example, in the spirit of Payton [Payton90] a navigational planner might generate a potential field representation of the navigation task, and this representation is used as a resource or suggestion by the “move-to” behavior when it determines the bearing to steer.

- **Will it scale?** While the action-selection and learning algorithm has been used to build one of the most sophisticated autonomous animated creatures to date, Silas’s repertoire is still extremely limited. Thus, it is a fair question to ask whether one could build a creature which was an order of magnitude more complicated than Silas using the same approach? Until it is done, this remains an open question. However, there are a number of aspects of the approach which suggest it will scale. First, through the explicit use of behavior groups and the implicit convention of using loose hierarchies, the interaction among behaviors is localized. This means that once a behavior sub-system is debugged, it typically isn’t perturbed by the addition of new sub-systems. The localization also means that there doesn’t necessarily have to be consistency across behavior groups as to the range for parameters such as the maxValues associated with Releasing Mechanisms. Second, the design of the motor system makes it straightforward to add new motor skills since the coordination among motor skills happens at the Degree of Freedom level. Third, the use of pronomes makes the behavior groups generalizable, e.g. one find-and-grab behavior group may suffice for whatever object a creature needed to find and grab. Lastly, one of the objectives of providing an integrated approach to learning is that it can be used to learn the appropriate values for key parameters.

Even if we are correct in our sense that the architecture is scalable, it is extremely important that better design and analysis tools be built so as to facilitate easier development, debugging and control of these creatures. As with most distributed systems it is often difficult to know just why the system is performing as it is. In our system this problem is compounded by the fact that it is such a rich model (although the use of Behavior Groups does tend to localize the interaction, and help in development and debugging).

- **Could one build a "human-like" character using this approach?** Characters with human-like qualities (e.g. an interactive Curious George) or caricatures of humans (e.g. an interactive Woody or Buzz Lightyear) are probably doable using this approach, assuming the context in which they are to perform is tightly bounded. The problem comes in the interaction with the user if the user is led to believe that the character is "smarter" or "understands" more than in fact it understands. With an animal it is easy to meet the user's expectations, with a broad caricatures of a human it is more difficult and the difficulty increases exponentially as it becomes less and less of a caricature. In any event, to do this we will need to understand how to integrate language and gesture production into the approach, and as mentioned above we would also probably need to integrate some level of planning. This would need to be done in conjunction with someone who intimately understood those issues. Ethology, though, has much to say about our own behavior so that many of its lessons are directly applicable to modeling caricatures of humans. This is a fascinating problem, and one which is well-worth doing.

7.0 Related Work

In this chapter we review related work, and put our work in context. The chapter is organized into three sections. The first section focuses on related work in action-selection. The second section focuses on learning. The last section focuses on related work in the area of autonomous animated characters.

7.1 Action-Selection

The problem of deciding what actions to take on a given instant is known as the problem of action selection. In the field of Artificial Intelligence there are two general approaches to this problem. The traditional approach views this problem in the context of constructing a plan to achieve a given goal. In this approach sensory data is used to construct a model of the world, and then on the basis of this model, a plan is constructed which specifies the actions required to achieve the desired goal from the current point. This plan thus specifies the appropriate action to take on a given step. Often planning systems try to find optimal paths to the goal. Shakey [Nilsson92] provides an example of how a planning system can be used in the context of a robot. While planning systems represent a powerful approach to certain classes of problems, there are several serious problems with them, particularly in the context of autonomous creatures. They do not do well in noisy and dynamic environments because the sensory input is often not sufficient to build a good model of the world. More importantly, plans assume that the world does not change in unexpected ways. If it does change, then the plan may no longer be valid. Indeed, it is often very difficult to determine how the world will change in response to a given action. In any event, the process of building a plan takes time and it is difficult for planning systems to exhibit real time performance. Planning systems also have difficulty balancing multiple competing goals and behaving opportunistically. Finally, planning systems tend to be very complex and difficult to implement and understand. For all of these reasons, traditional planning systems are not well suited to the problem of action selection in interactive animated creatures.

A number of researchers, led by researchers such as Brooks, Maes and Minsky argued that distributed collections of individually simple behaviors tied directly to the creature's sensors and actuators might in fact perform better than planning systems in dynamic, noisy, uncertain environments in which real time performance was a requirement. Their intuition, drawn from the work of ethologists, was that a great deal of animal behavior could be explained as the interaction of simple reactive behaviors without recourse to sophisticated planning abilities on the part of the animal. This school of thought was dubbed "Behavior-Based Artificial Intelligence", or BBAI. In contrast to planning systems, BBAI systems do not explicitly plan, and hence do not attempt to build models of the world. Change is not an issue in these systems because the system re-evaluates its course of action on every tick, so if the world changes it can respond accordingly. They are concerned with adequacy, robustness and opportunism rather than with optimality. The systems are often designed to take advantage of regularities in the particular environment for which they are intended so as to simplify the problem of perception. A premium is placed on real time performance. While all of the early work in BBAI focused on the problems associated with real robots, the problems they were attempting to solve and the approaches they took were similar to those discussed in this thesis. Here we will briefly review the work of several of the leaders in this field, including Brooks, Payton, Tyrrell and Maes and then put our work in the context of this work.

7.1.1 Brooks

Brooks [Brooks86, Brooks91] was one of the pioneers in the field of behavior-based AI and his subsumption architecture has had a profound impact on the field. In the subsumption architecture there is no central planner or controller. Rather it is a distributed collection of goal-directed entities, called behaviors which themselves are made up of collections of finite state machines. The key point in the architecture is that these behaviors run in parallel, and intelligent behavior emerges out of the interaction among behaviors. The behaviors themselves can inhibit other behaviors, or can subsume their functionality by over-riding them. In this manner it is possible to create a layered architecture. There are typically no explicit goals in a system built using this approach (although more recently the equivalent of a “hormonal system” was added to the architecture). However, goals are implicit in the design of the actual behaviors and in their allowed interaction. Sensory input is limited to bitfields. The architecture itself does not deal with the issue of how conflicting goals should be addressed or how external factors should be weighed against internal factors, or with the issues of insuring relevance and coherence in a system with multiple goals. In addition, identifying the appropriate set of behaviors and links for a given task is often a challenge. Command fusion (i.e. combining preferences for motor actions) is also not supported. Adding new behaviors can have global effects so it is often difficult to scale up.

The importance of the subsumption architecture for interactive characters is probably not in the specifics of its design, but rather in its role as motivating other work in the field. Tyrrell [Tyrrell93] argues that it is perhaps best thought of as a substrate upon which other architectures could be implemented, i.e. as a virtual machine.

7.1.2 Travers

Travers [Travers88] developed an early tool kit for building simple 2D autonomous creatures. The model was inspired both by ideas from ethology and Minsky’s Society of Mind [Minsky87]. His behavior system was composed of agents which he described as gated if-then rules. The condition part of the rule typically involved a sensory predicate and thus defined when the rule was applicable. The gate handled timing an activation strength (this system incorporated a model of spreading activation), and the rule defined the action of the agent. An agent could either issue motor commands or spread activation/inhibition to other agents. The model also incorporated a simple learning algorithm as well. A particularly interesting aspect of Travers’ work was his emphasis on agent-based languages as a useful programming paradigm.

7.1.3 Payton

Payton [Payton90,92] has developed several architectures for controlling autonomous robots. One common theme in his work has been an arbitration mechanism which allows behaviors to express multiple weighted preferences for motor actions (i.e. command fusion). In the simplest case, assume that the motor controller accepts 7 steering commands: full left, 2/3 left, 1/3 left, straight, 1/3 right, 2/3 right, full right. Rather than issuing a single turn command, a behavior may give its weighted preferences (positive or negative) for all seven commands. Behaviors are linked into a network and thus a behavior’s preferences are combined with the preferences of other behaviors and the command with the greatest weight wins. The fundamental problem with this approach is that the weight used by a given behavior for a given action can not be set independent of those used by other behaviors. This means that the weights have to be carefully chosen, and the addition of a new behavior may necessitate re-adjustment of the weights. Tyrrell [Tyrrell93] also argued that care needs to be taken with respect to how the weights are combined. For example, many behaviors may care only a little about moving in a certain direction, but because there are so many of them they may swamp the desires of a single behavior which cares a great deal about moving in another direction.

In a later paper [Payton92], Payton focused on a somewhat different problem, namely insuring robust autonomous behavior in environments in which it was highly probable that the system would encounter events that could not be anticipated at design time (e.g. an autonomous submarine). His observation was that it was not necessary for the system to figure out why things weren't going as planned. All the system had to do was to recognize that a behavior was not having the intended effect, and switch to an alternative behavior which accomplished the same task but in a different manner. Specifically, every behavior should be capable of monitoring its own effectiveness at producing its desired outcome, and be capable of adjusting its actions accordingly, including the option of lowering its priority so that a behavior with a competing strategy can take control. For this to happen, a behavior needs visibility into the effect of its actions, and a way of modulating its strength vis-a-vis competing behaviors as appropriate. In a system in which multiple behaviors may express preferences, the behavior also needs to know its relative contribution to the actual actions performed, or what Payton calls its "participation".

In this later model, behaviors are organized into overlapping collections of compatible behaviors. That is, within a given collection all of the behaviors may run concurrently without competing. However, the collections themselves compete. Another interesting idea from Payton's work is his use of alternative methods of specifying commands. Rather than specifying preferences as discrete outputs, he proposed using alternative forms such as "Spikes" which correspond a specific control setting, "Zones" which correspond to a control settings within a given range, and "Clamps" which place lower or upper bounds on the desired control setting.

Due to vagaries of government funding, the system was never actually implemented. Nonetheless, there are a number of very interesting ideas in his work. As we noted earlier in section 5.2 we extended Ludlow's notion of Level Of Interest to reflect Payton's "progress-based" approach.

7.1.4 Tyrrell

Tyrrell [Tyrrell93] like ourselves was influenced by ideas from ethology and incorporated Payton's approach into an ethologically inspired model and reported good results. One improvement he made was to note that a simple sum is not a good way of combining preferences for a given control setting (i.e. full-left above corresponds to a control setting). For example, if many behaviors care just a little about going right, but one behavior cares a great deal about going left, the result may be to turn right even though it might make more sense to turn left. Thus, he proposed a variant in which the preferences for a given control setting should be calculated as a weighted average of the highest single preference and the sum of all preferences for that setting.

One disadvantage of Payton and Tyrrell's approaches, and indeed one which is inherent in any voting scheme in which a behavior expresses a weighted preference for a given action, is that the weights need to be adjusted carefully and in concert. Adding a new behavior may require modifications to existing weights.

7.1.5 Maes

The systems of Brooks, Payton and Tyrrell all involved “hand-built” networks, and with the exception of Tyrrell minimized the importance of explicit goals. By contrast, Maes proposed an architecture with explicit time-varying goals. In Maes' model, there is no hierarchy (although this is not precluded), but rather the behaviors are organized in a network with causal links connecting the behaviors. That is, every behavior has a set of pre-conditions which must be satisfied in order for the behavior to become executable, and a set of post-conditions which it promises will be true after it has become active. These pre and post conditions are used to form causal links among the behaviors. Behaviors use these causal links to spread activation energy to other behaviors and in effect recruit these behaviors to perform some action which helps achieve the first behavior's goal. So for example, the “reduce-hunger” goal recruits the “chewing” behavior which in turn recruits “move to food” which in turn recruits “find food” if food isn't visible, and so on. This is in contrast to the traditional hierarchical approach in which “hunger” would activate the “feeding subsystem”, which in turn would invoke “find and move to food” if food wasn't present, or “chew” if it was. For a discussion of the advantages and disadvantages of each approach see [Blumberg94, Tyrrell93, Maes90]. Maes’ algorithm is unique among the behavior-based approaches in its ability to do simple planning.

Tyrrell [Tyrrell93] has criticized Maes' overall action-selection algorithm on a number of grounds. For example, her system as described has an impoverished model of perception (all states in the world are represented via predicates), there is no mechanism for combining preferences, and there is little support for controlling the temporal aspects of behavior. While these may be valid criticisms of her algorithm as a complete solution to action-selection, it should be noted that these criticisms are not fundamental to her approach (i.e. mechanisms could be added to address those concerns). Indeed, Rhodes [Rhodes96] has modified the Maes system to incorporate many of these features. Moreover, they are orthogonal to her major contribution of showing how a spreading activation model could be used to implement a simple form of planning.

Maes has also implemented a very elegant extension to her algorithm [Maes90c] to allow the behaviors to learn the causal links which connect them. In earlier versions of her system, these links were hardwired by the programmer. In her extension, the behaviors keep statistics on the state of their pre-conditions. Whenever a pre-condition's state changes, the behavior attributes responsibility to the last active behavior, and if no link exists, it will create one, and if one exists it will strengthen it (links have a weight which indicates the “reliability” of the link a/b, where a = number of times that condition i became true and activity j was active, and b = total number of times that condition i became true). This technique was used to teach a six-legged robot to walk [Maes90c].

7.1.6 Summary

Our action-selection algorithm differs from previous models in several important respects. First, our treatment of external and internal factors as first class objects distinquishes our approach from that of Maes [Maes90a], Brooks [Brooks86,91], Rosenblatt[Rosenblatt89], Tyrrell [Tyrrell93], and Tu[Tu94] which typically represent external factors as predicates. Second, our system is the only system which provides explicit mechanisms for controlling persistence. Our model of arbitration and use of inhibitory gains was taken from Ludlow [Ludlow76,80], although we use it in a context which is closer in spirit to that proposed by Minsky [Minsky88] (our behavior groups are equivalent to cross-exclusion groups). Level of Interest is taken from Ludlow as well, but we have broadened the concept (in Ludlow level of interest is largely a temporal construct) to provide a generalized mechanism for controlling persistence based on the behavior's own sense of its progress. Payton's [Payton92] architecture incorporates similar ideas, both in terms of organizing behaviors into competing groups, and the use of a feedback mechanism to modulate the value of a behavior.

The concept of releasing mechanisms is directly taken from ethology, and represents a novel approach to assessing the relevance of the environment for a given behavior. Other approaches do not provide such a rich model. Booker's [Booker88]Gofer system did, however, incorporate Releasing Mechanisms, although they were much simpler.

Our use of pronomes is novel, although the idea, of course, is taken from Minsky [Minsky88], and Rhodes had previously suggested that pronomes might be a useful addition to Maes algorithm [Rhodes95].

Our approach toward allowing multiple behaviors to express their preferences for motor actions, and the underlying support provided by the motor system to blend, modulate and provide concurrent execution of compatible motor actions is novel. The action-selection algorithms of Maes [Maes90a] and Tu [Tu94] do not provide support for allowing multiple behaviors to simultaneously express their preferences for motor actions. The algorithms of Tyrrell [Tyrrell93] and of Rosenblatt and Payton [Rosenblatt89,Payton93] do support command fusion, but this comes at the expense of careful tuning of weights, and the potential that all of the weights may need to be adjusted if a new behavior is added to the system. Furthermore, Tyrrell's use of this mechanism was limited to low-level navigation. Our use of multiple imperative forms for motor commands is novel, as is our design for the underlying components of the motor system. Perlin [Perlin95] has also investigated how to combine and blend motor actions, but he has not integrated this into an overall architecture for action-selection and learning.

7.2 Learning

There has been an enormous amount of work done in the area of learning. However, much of it is very different in flavor from our work. Thus, rather than review the approaches of specific researcher we will review two broad trends: reinforcement learning and model learning.

7.2.1 Work in Reinforcement Learning

There has been no shortage of work in the area of reinforcement learning for agents. See [Baird93, Kaelbling92, Klopf93, Lin91/92, Mahadevan91, Sutton90/91, Whitehead92, Watkins89, Wilson91, Holland86] for examples of this approach. In reinforcement learning the agent attempts to learn an optimal policy to achieve a given goal. A policy is nothing more than a decision rule which, in effect, says “If in state X, do Y”. While there are a number of ways to skin this cat, much of the work is derived from that of Watkins who developed the Q learning algorithm and showed that it would produce an optimal policy in a world in which the creature could visit every state, and in which every state could be unequivocally identified. Q learning in turn is really just an iterative form of dynamic programming. Think of it as dynamic programming for worlds in which you don’t know the state transition matrix.

Unfortunately, there are a number of problems which have proved very difficult for reinforcement learning. A fundamental problem is the so-called curse of dimensionality. If the state space is at all large, and if the set of possible actions is large as well, it may be difficult or impossible to explore all of the state space. Thus, it is critical to find ways to limit the search space, since after all vast portions of the state space are probably irrelevant. Kaebling among others has looked at this problem. Another serious problem is that of perceptual aliasing in which one state can not be disambiguated from another. This can be a particular problem if the system incorporates focus of attention.Whitehead and Lin have looked at this issue and offered some possible strategies. For example, Lin suggests the use of temporal windows, in which the state information is augmented by some history. Learning occurs on the fringes because, in effect, value flows back from the goal. Thus, no learning occurs (other than perhaps learning the state transition matrix, see [Sutton91]) until the goal is achieved. Even then, it can take many trials to propagate the value back in time. There are issues with how the system should explore the space and when it should experiment vs. exploit a known path. The learning is relative to a given goal. If the goal changes, or the world changes, then everything needs to be relearned.

Note, I have included classifier systems [Holland86, Wilson91, Wilson96] in with reinforcement learning because while the exploration strategy may be different, they are both trying to learn a value function and ultimately an optimal policy.

McCullum [McCullum96] has developed an approach to the perceptual aliasing problem in which he combines memory-based learning with statistical techniques. Fundamentally, his system keeps a “tape” of its entire experience, and uses this tape to learn how much of its history it needs to pay attention to in order to disambiguate between states. The interesting aspect of the algorithm is that it learns to make finer distinctions (i.e. go back further in time) in certain portions of the state space than in other parts of its state space. This dramatically reduces the size of the state space. For example, in his problem domain this approach reduced the size of the relevant state space from 10^{10} states to 143 internal states.

Most recently, Humphrys [Humphrys96] has done some very interesting work in which he uses reinforcement learning to both learn the optimal set of actions for a given task, and to decide on which task to work at any given instant. While standard Q-learning is used to learn how to perform each task (the state space associated with the individual tasks is small enough that table-based Q learning works), a different technique is used for the arbitration among tasks. Specifically, the system chooses the task which would suffer the most if it weren’t chosen, and this is factored back into the learning system so that the agents responsible for the individual tasks learn, in effect, how to compromise.

While he has several variations on this approach the important thing is that it produces a comparable or better solution than hierarchical Q learning and in far fewer steps.

7.2.2 Model based approaches

Drescher[Drescher91] represents an alternative school. He is not concerned with learning an optimal policy but rather in understanding the world, and in particular, reliable and relevant correlations. In other words, he is interested in building a model of the world which can be used to guide action. His system is elegant in its conceptual simplicity. Essentially the system keeps statistics on how the world changes when it takes a given action. If it notices that a particular action appears to be relevant to a given outcome (i.e. if it appears that it is more likely that an outcome will occur if an action is taken, than if the action isn't taken), it then tries to learn the context in which performing the action will more reliably lead to the outcome, than performing it in other contexts. In this manner the system builds up "schemas" which map a context and an action into an outcome.

There are many clever ideas in Drescher's approach and he used it to demonstrate certain forms of Piagetian learning. The weakness in his original approach was that there was no focus of attention, either spatially or temporally. Foner and Maes [Foner94] modified the algorithm to incorporate focus of attention and were able to report significant performance gains (30 times in some cases!!!). This was a good thing, as Drescher's original algorithm could bring a Connection Machine to its knees on a very simple 7x7 grid world.

Touretsky and Saksida [Touretsky96] have developed a model-based approach which is similar to ours in spirit in that they focus on developing a model for operant conditioning based on results from animal behavior. The system learns the reliability of conjunctions of stimulus, actions and outcomes, and uses this information to guide action. Temporal contiguity is used to provide focus of attention. They have tested the system with a delayed-match-to-sample experiment.

Corbacho and Arbib [Corbacho96] have also investigated an approach in which the system learns useful correlations between stimulus, actions and outcomes. As in our system, the goal is to learn how to use old behaviors in new contexts, and how to chain schemas. The fundamental idea in their system is that an animal has an innate set of schemas (i.e. combinations of stimulus and behavior) and a set of expectations about the results of performing those schemas. On the theory that usually the expectations match what in fact happens, the system focuses on those relatively few situations in which something unexpected happens. When this happens the system builds a new prediction and over time learns a weight to place on this prediction.

Maes [Maes90b] used similar ideas of tracking the relevance and reliability of correlations in the world to integrate learning into her action selection algorithm. Indeed, she was in having a robot insect learn how to walk using her approach.Giszter [Giszter94] also used an approach based on Maes' algorithm to perform motor learning in a simulated frog.

7.2.3 Other Work

A number of researchers have proposed computational models of classical and operant conditioning for animats. These include Sutton and Barto [Sutton90], Klopf [Klopf93], Baird[Baird93] and Schmajuk [Schmajuk93,Schmajuk94]. These researchers have tended to focus on modeling the details of classical or operant conditioning experiments, and are less concerned with the issue of learning within the context of an existing behavior system, although Klopf demonstrated maze learning with his system and Schmajuk demonstrated both place learning as well as escape and avoidance behavior with his. Schmajuk [Schmajuk93, Schmajuk94] and Balkenius [Balkenius96] have also looked at the issue of generalization across stimuli in the context of operant conditioning.

Vershure [Vershure91] used a simple form of classical conditioning to teach a robot to take anticipatory actions to avoid collisions with obstacles. This was a particularly clear example of how one might integrate learning into an already existing behavior system.

Halperin [Halperin91] proposed a learning model, based largely on hebbian learning, in which the system would learn sensory-motor interactions which addressed three kinds of different motivations. The first kind of motivation corresponds to "drive-reduction" motivations such as "hunger". The second kind corresponds to "incentive" or hedonic motivations (i.e. "pain" or "pleasure"). The third is what she refers to the motivation to perform sequences of motor actions. She shows how a simple neural net model could, in principle learn stimulus behavior pairs which addressed each of these different types of motivations. The system was used to model "Siamese Fighting Fish" social behavior.

Todd and Miller [Todd91] looked at the relationship between evolution and associative learning and discovered the interesting U-curve phenomena. In there experiment a creature had a built-in predictor of reinforcement which was of varying reliability. They were interested in the difficulty of evolving another more reliable predictor. Essentially they found that associative learning evolved most rapidly when the built-in predictor of moderate accuracy. If it was too accurate, associative learning wouldn't evolve because there was no selective pressure. On the other hand, if it was too inaccurate it would also not evolve because it was too difficult to learn. The larger point of the exercise was to point out that "useful feedback" doesn't magically happen, but evolves as well. This evolution of feedback, in turn, is an essential part of the learning process.

7.2.4 Summary

Our approach to how we integrate learning is very much inspired by the ideas of Lorenz [Lorenz73], Hogan [Hogan88], Shettleworth [Shettleworth94] and Dickinson [Dickinson94]. The underlying TD learning model is taken from Sutton and Barto [Sutton90], and of course TD learning itself is a very popular learning technique [Klopf93, Kaebling92, Lin91/92, Mahadevan91, Sutton91, Watkins89, Whitehead92]. However, we are less interested in learning an optimal policy for an extended delay reinforcement situation. Our contribution is more to show how TD learning can be integrated into ethologically inspired action selection mechanism. Our memory model is taken from Killeen [Killeen94], and our use of focus of attention is similar to that proposed by Foner and Maes [Foner94], although our view of relevant contiguity is different (i.e. not strictly spatial or temporal).

As mentioned earlier, our integration of learning and action-selection sets us apart from most previous work in the field (the notable exception being Maes [Maes90]). For example, the action-selection algorithms of Tyrrell [Tyrrell93], Tu [Tu94], Brooks [Brooks87], and Payton [Payton92] do not incorporate learning (although Payton's system does learn via feedback whether a particular action is having the desired result). Similarly, the research in learning typically focuses on learning an optimal policy for a single goal and thus does not address the more general action-selection problem.

Our approach to learning differs from Maes [Maes90] in that we integrate TD learning whereas Maes essentially used correlations to learn links between competence modules and relevant conditions. Her system produces a form of instrumental learning, but does not demonstrate many of the phenomena associated with classical conditioning (e.g. blocking or second order conditioning), or operant conditioning (e.g. effects of variable reinforcement). Maes approach does not learn how long to engage in a given behavior, whereas our system also learns the time course associated with behaviors which act as appetitive behaviors.

Our approach differs from Touretsky and Saksida [Touretsky96] in that in their system the learning isn't integrated into an existing behavior system, and there is no attempt to learn a value to be associated with a given stimulus-behavior pair.

More generally, our work differs from most work in machine learning in that our emphasis is on using a simple learning mechanism embedded in pre-existing and rich structure. This is a very different approach than, for example, Drescher [Drescher91], who presumes learning takes place in a *tabla rasa* environment.

7.3 Autonomous Animated Characters

There have been a number of impressive efforts in the area of behavior-based animation. Here we highlight a few of the most important examples of this work.

7.3.1 Reynolds

Reynolds was one of the first to investigate behavioral control of animation. He developed a system to model the flocking characteristics of birds and fish. His observation was that global flocking behavior could emerge from the application of simple local rules by each of the flock members.The behavior system for each boid needed to arbitrate among 4 possibly conflicting goals: (1) Avoid collisions with nearby boids, (2) Match velocity with nearby boids, (3) Stay close to nearby boids, (4) Attempt to fly toward a goal. Arbitration was done using a fixed priority scheme. Each boid had a maximum allowed acceleration. The desired acceleration from each desire was computed and added in priority order until the magnitude of the resulting acceleration exceeded the maximum allowed acceleration. The priority order could be varied at runtime depending on environmental conditions. His behavior system is focused on two specific goals, namely allowing the boid to avoid "painful collisions" and fly to a given point, and thus it was not intended as a general behavioral model. Nonetheless it represents a good example of blending the preferences of multiple low-level behaviors to arrive at a compromise solution.

Reynolds also pioneered the use of synthetic vision. Reynold's boids incorporated a steer-to-avoid model for collision avoidance whereby the boid finds the closest silhouette edge to the point of contact (the center of the visual field), and aims itself to pass by the edge with a suitable tolerance.

It is probably worth noting that of all of the systems discussed in this thesis, Reynold’s work has been seen by the most people! This is because Reynold’s boid model is commonly used today in Hollywood for such special effects as the bats in “Batman”, and the crowd scenes in the “Hunchback of Notre Dame”.

7.3.2 Tu and Terzopoulos

Tu and Terzopoulos [Tu95], in an impressive effort have implemented a physically-based artificial fish model which also incorporates a perceptual system and behavior system. Unlike Reynold's boids, the fish have a fairly complicated physical model which allows the fish the generate realistic looking motion. The behavior system is also more complicated than that of Reynolds and allows the fish to engage in a number of high level activities such as feeding, mating, schooling.

The behavior system is designed around an “intention generator” which is essentially a fixed and prioritized set of rules which determines what primitive behavior is executed. The primitive behaviors include: avoiding static obstacle, avoiding fish, eating-food, mating, leaving, wandering, escaping, and schooling. The rules take the form of predicates where the terms may be either sensory data (distance to closest predator) or internal state variables (fear threshold). A simple 1 element stack is provided to allow the intention generator to “remember” interrupted intentions (for example, if it is engaged in mating and it must avoid an obstacle, it will push mating and the its current mating target onto the stack). This is done to provide some measure of persistence. Another mechanism called a “focusser” is provided which extracts relevant information from the sensory system for the low level behaviors. The focusser is made active once the intention generator has determined which behavior should be active, so it presumably chooses which information to extract from the world based on that knowledge.

Ultimately, the primitive behaviors may issue one of 3 motor commands: Swim, Turn Left and Turn-Right.The motor controller takes these commands and maps them into control signals for the simulated muscles of the fish. A particularly interesting aspect of the system is that the fish “learn” how to swim using optimization techniques which attempt to optimize efficiency of its swimming subject to certain constraints. Most recently he has used similar techniques to teach his fish to perform the kinds of tricks one sees in marine animal parks [Terzopoulos96]

Terzopoulos’ fish also use synthetic vision for sensing. The eyes of his fish are implemented with multiple virtual cameras which approximate both peripheral and foveal vision. Active vision techniques are used to detect and foveate on a target, with color indexing used to actually identify a given target. By mapping the actions necessary to visually track a target into the appropriate control signals, his fish are capable of visually guided navigation.

The behavioral model, while producing realistic fish behavior has a number of important limitations:

- There is no mechanism to explicitly control persistence among competing goals. Thus, the system may be subject to dithering and pathological persistence which they have been able to avoid up to now by having a fixed hierarchy of needs.
- It uses a fixed priority scheme with fixed thresholds. Avoid always dominates avoiding predators which always dominates eating and mating. While a simple fixed hierarchy may suffice in this situation, as the number of behaviors grows, it is much less likely to. For example, the threshold for obstacle avoidance is much different if the creature is approaching the object for the purpose of stopping near it, then it is if it is moving quickly and wants to insure that it doesn't hit the object.
- Scaling will be a problem. Adding a new behavior means modifying the intention generator, changing the focusser, and adding the primitive behavior. Similarly, changes to a behavior, or the addition of a motivational variable may in turn necessitate changes in the intention generator and focusser.
- There is no mixing of preferences, so the system is not capable of arriving at compromise solutions. Nor is there a mechanism to allow the creature to act opportunistically. In general, the system does not address the problem of doing multiple things at one time.

7.3.3 Bates

One of the earliest, and best known efforts is that of Bates [Bates92,Bates94]. Bate's work is similar to ours in that his creatures, Woggles, attempt to juggle multiple goals based on their perception of the world and their internal state. Indeed, his creatures have an elaborate emotional model. He also places great emphasis on conveying motivational state and intentionality through movement[30]. However, his system differs from ours in a number of ways. First, his system is largely rule-based rather than the distributed approach advocated here. Nor is learning incorporated into his model. Interaction with his characters is limited to mouse movements and clicks. It is unclear how extensible his architecture is: woggles are the primary instance of it, and the world in which the woggles inhabit is very constrained. He has also made life very difficult for himself by choosing "ellipsoids with behavior" as his vehicle for displaying the results of his elaborate architecture. To convey what he claims his system is capable of conveying through woggles would be a challenge for a world-class animator to say nothing of a graduate student.

7.3.4 Perlin

The Improv system of Perlin and Goldberg is designed as a tool for animators to create interactive actors [Perlin96,Perlin95]. They are less concerned with problems of autonomous behavior than they are in providing a powerful way to script behavior. The design of the motor system is somewhat similar to ours in that they utilize many of the same concepts (Degrees of Freedom, Motor Skills). However, they add additional structure in that motor skills are grouped into competing motor skills, and groups of motor skills are ordered back to front with more global motor skills in back and more localized skills in front. They use a method analogous to compositing to blend the result of motor actions. This, coupled with some other interdependencies among motor skills, means that they

30. If we are inspired by ethology and classical animation, Bates is more inspired by traditional AI and classical animation.

can do smooth transitions. He also incorporates a noise function. In short, their approach to motor control is very interesting and compatible with our overall approach so we may move to incorporate some of their ideas in future versions of the system.

At the behavior level, they rely heavily on the scripting abilities of the animator where scripts are essentially sets of decision rules which govern the character's behavior. There are no explicit mechanisms for insuring relevance, coherence or opportunistic behavior. Similarly, there is no mechanism for incorporating learning. This reflects their view that they are building "actors" not "creatures". Perlin would argue, I think, that it is much easier to build an actor who can look "sad" on command, than it is to build a creature who looks "sad" because he *is* "sad". While it is hard to debate this point, the real question is how well the actor model will work in unconstrained and noisy environments. The danger, we think, is that because the actors do not have "behavioral integrity", they are prone to inconsistent or inappropriate behavior as the environment becomes less constrained and the animator has less ability to predict all of the possible outcomes.

7.3.5 Dogz

Dogz [PF.Magic96] is a very appealing commercial product in which the user "adopts" an autonomous animated puppy. Over time the puppy matures into a full grown dog. The interaction with the user is rich: the user may feed and water the dog, pat and scratch him, play fetch and tug-of-war, and even train it to do tricks in return for rewards. The dog sleeps, howls or wanders around the screen if the user isn't playing with it. There is even a screen saver mode in which the dog becomes a watch-dog, barking if someone touches the keyboard without giving the correct password. The 2 1/2-D animation is very cartoonish but reasonably effective. The animation incorporate a simple noise model, as every part of the dog's body seems to be in motion all of the time.

As entertainment, Dogz works rather well. It is difficult to know what is under the hood, although the learning model is clearly much simpler than that proposed here (neither chaining nor classical conditioning is supported). Nonetheless, it is an impressive effort.

7.3.6 Others

Fisher's Menagerie system [Fisher93], Tosa's "Neuro-Baby" [Tosa93], and Maudlin's "Julia" [Maudlin94] are additional examples of systems in which the user can interact with autonomous interactive characters. Badler [Badler93] has done impressive work in bio-mechanical modeling of the human body, largely for human factors analysis. He has also explored task planning. His system served as the underlying technology for Cassell's work in animated conversation.

7.3.7 Summary

Our work is distinguished from this work in a number of ways. First, none of the systems with the exception of Dogz incorporates learning at the behavioral level, and even in Dogz the learning is extremely limited. Second, in contrast to Reynolds, Terzopoulos, Bates and Perlin, our system is the only one which provides a general behavior model. Reynolds focused on only a single behavior, i.e. flocking, Terzopoulos' model is intertwined with the specifics of the fish behavior he wished to model and the specifics of his motor model. None of the models provide explicit mechanisms for insuring relevance or controlling persistence. Third, while Perlin's system does emphasize the need for concurrent and coherent motor action, the other systems do not address this issue. However, Perlin is less interested in the question of generating intelligent behavior, than in the question of what tools animators need so that they can create scripts which create the illusion of intelligent behavior. Bates' system, while undoubtedly incorporating a rich emotional model has been hampered by their choice of "Woggles" as a vehicle. Fourth, with the exception of Perlin's system, all of the other systems have been used to build a single type of creature for a specific environment. By contrast we have built a variety of creatures for several different environments including the ALIVE environment.

8.0 Concluding Remarks

8.1 Areas for future work

There are a number of important issues which are outside the scope of the work presented in the thesis, but which we would like to pursue in the future. We list some of most important issues here:

- **Explicit model of affect and emotion**. Silas has a very rudimentary emotional model, and while we believe our model of motivational factors provides a rich mechanism with which to implement an explicit model of affect, our system does not currently incorporate one. However, we are confident that the models proposed by researchers such as LeDoux [LeDoux90] or Damasio (at least with respect to primary emotions) [Damasio95] could be integrated into the architecture described here.
- **A "Theory of Mind"**. What has been discussed in this thesis is fundamentally mechanism. We have not presented a larger theory of mind. For example, what representations, if any, are necessary beyond Releasing Mechanisms? Or, are there identifiable "mental states" and clear transitions between them? Or, how does a creature sense the motivational state of another creature? Indeed, such a theory of mind will ultimately be necessary to incorporate affect and emotion into our model.
- **Motor Learning.** Our approach to learning does not address learning new motor skills and assumes the set of motor skills is fixed. See Giszter [Giszter94] for an example of motor learning using an algorithm derived from Maes' algorithm [Maes90a]. Also see Terzopoulos [Terzopoulos96] for another example of motor learning. In particular, we would like to explore motor learning using motion generated by animators as the basis for the learning. Here an animator might generate a sequence of movements and the system would learn a model from which it could interpolate or extrapolate from the animator-provided motion.
- **Generalization**. Our learning algorithm does not address the problem of generalization across stimuli.
- **Extended delay reinforcement and optimality.** Although our use of a temporal difference model is closely allied with approaches such as Q-learning, no claim is made that our approach will find optimal solutions or is equivalent to Q-learning. But see Baird [Baird93] for a discussion of how the Klopf algorithm (which is related to the underlying learning algorithm used in our system) relates to Q-learning
- **Planning.** There is no explicit planning capability in our system. For example, all navigation is reactive without recourse to internal representations of the world such as remembered landmarks, or "cognitive maps". Animal navigation is a particularly rich area from which to draw ideas for autonomous creature navigation. However, we do not address these issues in this thesis other than to note that many of the mechanisms proposed by Wehner (path-integration); Srivansian, Lehner and Collett95 (visually-guided navigation); and McNaughton (role of the hippocampus in navigation) [Wehner95, Srivansan95, Lehner95, McNaughton95] would appear to be both useful in our domain as well as computationally feasible to implement. See Mataric [Mataric90] for an example of a landmark-based approach to navigation inspired by McNaughton's earlier work on the hippocampus.

- **Exploration and experimentation.** In the current learning system the creature does not bias its exploratory behavior based on preliminary feedback, and this should be changed. For example, if Silas noticed that he was more likely to receive a reward when he sat then when he didn't, he should then bias his choice of actions toward sitting, if for no other reason than to give him more of a chance to learn the context in which sitting is a reliable strategy for getting food. This would be a simple and useful change.

  In general, we would like to explore how we can make it easy for a user to train these creatures.
- **Extend our use of synthetic vision.** We would like to increase our use of synthetic vision, perhaps incorporating some of Terzopoulos' ideas [Terzopoulos96]. This is important because believable perception is essential to believable behavior.
- **More life-like motion.** The current motor system does not incorporate a noise model such as that proposed by Perlin [Perlin96], and probably places too much emphasis on making motor skills independent of each other. As noted earlier, this makes transitions less natural than they should be. Thus, in the future we would like to incorporate a noise function, and move to an approach that allows more arbitrary transitions from one Motor Skill to the next.

8.2 Conclusion

Despite the initial enthusiasm and claims made for behavior-based AI, there are probably those who feel that the approach has not lived up to its supposed potential. The argument presented in this thesis is that the original vision was correct in that biology, and in particular ethology, could offer valuable insights into how to build intelligent creatures. If researchers erred, they did so by not paying enough attention to the lessons of ethology.

In this thesis we have examined three traditionally hard problems for autonomous agents in general and by taking ideas from ethology have presented a computational model of action-selection and learning for autonomous agents which improves on previous efforts. Specifically, there are two major contributions of this aspect of the work:

- An ethologically inspired model of action-selection and motor control for autonomous animated creatures which by explicitly addressing the issues of relevance (i.e. "do the right things") and coherence (i.e. "show the right amount of persistence") improves on previous work in action-selection for multi-goal autonomous agents.
- An ethologically inspired approach to learning which by incorporating temporal-difference learning into this architecture allows the agent to achieve its goals either in new ways or more efficiently and robustly.

In addition, we examined two issues of particular relevance to the problem of building autonomous animated creatures, namely the need to convey motivational state and intentionality through concurrent motor actions, and the need to integrate real time external control at multiple levels of abstraction. There are two major contributions of this aspect of the work:

- A motor system architecture which provides the necessary level of concurrent, coherent motion, and which allows multiple behaviors to express their preferences for motor actions. This approach not only facilitates the display of motivational state and intentionality but it also addresses a computational concern associated with the hierarchical organization of behaviors as proposed by ethologists.
- We show how our approach lends itself to the straightforward integration of external control at multiple levels of abstraction.

We have implemented this architecture in the Hamsterdam tool kit and have used it to develop a number of robust working examples of autonomous animated creatures including Silas, a virtual animated dog, which is one of the most sophisticated autonomous animated creatures built to date.

At a higher level, we hope that this work will contribute to providing a less ad-hoc basis for the field of behavioral animation. Previous work has focused on specific applications (e.g. flocking birds), or specific problems (e.g. locomotion). By contrast, we have presented what we see as the five key issues associated with building interactive animated creatures (i.e. Relevance, Persistence and Coherence, Intentionality, Adaptation and Integration of External Control) and have described and implemented an approach which addresses these problems. While there may be other issues which need to be added, and alternate approaches, we believe this work makes a significant contribution toward framing the problem and ultimately toward understanding how to build characters for interactive environments, be they virtual companions in immersive story-telling environments, or interesting and adaptive opponents in interactive games, or as the basis for “smart avatars” for web-based worlds.

While our focus has been on the problem of building autonomous animated creatures, we have argued that the importance of this work extends beyond this specific domain. From the standpoint of science, these creatures provide a model for exploring the issues associated with building intelligent systems which have multiple goals, which must convey their intentionality and which must function in real-time environments. Ultimately, they may also serve as a vehicle to better understand our own minds.

9.0 References

Arkin, R. (1990). Integrating Behavioral, Perceptual and World Knowledge in Reactive Navigation. In: *Designing Autonomous Agents*. P. Maes, ed. MIT Press, Cambridge.

Azarbayejani, A., C. Wren and A. Pentland (1996) *Real-time 3-D tracking of the human body.* In: Proceedings of the 3rd International Conference on Communicating by Image and Multimedia, 1996.

Badler, N.I., C. Phillips, and B.L. Webber (1993). *Simulating Humans: Computer Graphics, Animation, and Control*. Oxford University Press, New York.

Baerends, G. (1976) On drive, conflict and instinct, and the functional organization of behavior. In: *Perspectives in Brain Research 45*, 1976.

Balkenius, C.(1996). Generalization in Instrumental Learning. In: *From Animals To Animats, Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior*, September 1996, MIT Press. Cambridge Ma

Baird, L. and A. Klopf (1993). A Hierarchical Network of Provably Optimal Learning Control Systems: Extensions of the Associative Control Process (ACP) Network. *Adaptive Behavior,* Vol. 1, No. 3.

Barker, L.M. (1994). Learning and Behavior: A Psychobiological Perspective. MacMillan College Publishing Co., Inc. New York, N.Y.

Bates, J. (1994). *The role of emotion in believable characters.* In: Communications of the ACM, 37,7.

Bates, J. and A. Loyall, W. Reilly (1992) *Integrating Reactivity, Goals and Emotions in a Broad Agent.* In: Proceedings of the 14th Annual Conference of the Cognitive Science Society, Indiana, July 1992.

Bekoff, M. and D. Jamieson, ed. (1996) *Readings in Animal Cognition.* MIT Press, Cambridge Ma.

Blumberg, B. (1994). Action-Selection in Hamsterdam: Lessons from Ethology. In: *From Animals To Animats, Proceedings of the Third International Conference on the Simulation of Adaptive Behavior*, Cliff, D., P. Husbands, J.A. Meyer, and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Blumberg, B. and T. Galyean (1995). Multi-level Direction of Autonomous Creatures for Real-Time Virtual Environments. In: *Proceedings of SIGGRAPH 95*.

Blumberg, B., P. Todd and P.Maes(1996). No Bad Dogs: Ethological Lessons for Learning. To be published in: *From Animals To Animats, Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior*, September 1996, MIT Press. Cambridge Ma.

Bonner, J.T. (1980). *The Evolution of Culture in Animals*. Princeton University Press, Princeton N.J.

Booker L. (1988). Classifier Systems that Learn Internal World Models. *Machine Learning Journal*, Volume 1, Number 2,3.

Braitenberg, V. (1984) *Vehicles: Experiments in Synthetic Psychology.* MIT Press, Cambridge Ma.

Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation* RA-2.

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

Bruderlin, Armin and T. W. Calvert (1989). *Dynamic Animation of Human Walking.* In: Proceedings of SIGGRAPH 89.

Cassell, J, Pelachaud, C., Badler, N., et al (1994) Animated Conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In: *Proceedings of SIGGRAPH 94.*

Chapman D. (1991) *Vision, Instruction and Action*. MIT Press, Cambridge Ma.

Cheney, D.L and R. M. Seyfarth (1990) *How Monkeys see the world.* University of Chicago

Press., Chicago.

Cheng, P. and K.J. Holyoak (1995). Complex Adaptive Systems as Intuitive Statisticians: Causality, Contingency, and Prediction. In: *Comparative Approaches to Cognitive Science,* Roitblat, H.L. and J.A. Meyer, eds. MIT Press, Cambridge Ma.

Corbacho, F.J. and M. Arbib(1996). Learning to Detour & Schema-based Learning. In: *From Animals To Animats, Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior*, September 1996, MIT Press. Cambridge Ma

Churchland, P.S., V.S. Ramachandran, and T.J. Sejnowski (1994). A Critique of Pure Vision. In: *Large-Scale Neuronal Theories of the Brain.* Koch, C., and J.L. Davis, eds. MIT Press, Cambridge Ma.

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

Dawkins, R. (1976). Hierarchical Organization: A Candidate principle for ethology. In: *Growing Points in Ethology.* Bateson P. & Hinde R. ed. Cambridge University Press.

Davey G. (1989). *Ecological Learning Theory.* Routledge Inc., London.

Dennett D. (1987). *The Intentional Stance*. MIT Press, Cambridge Ma.

Dennett D. (1996) *Kinds of Minds.* Basic Books, New York, NY.

Dickinson, A. (1994). Instrumental Conditioning. In: *Animal Learning and Cognition,* Mackintosh, N.J. ed. Academic Press, San Diego.

Drescher, G.L.(1991) *Made-Up Minds: A Constructivist Approach to Artificial Intelligence.* MIT Press, Cambridge.

Duchon, A.(1996). Maze Navigation Using Optical Flow. To be published in: *From Animals To Animats, Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior*, September 1996, MIT Press. Cambridge Ma.

Fisher, S. et al.(1993) *Menagerie*. In: Siggraph-93 Visual Proceedings, Siggraph 93.

Foner, L.N. and P. Maes (1994). Paying Attention to What's Important: Using Focus of Attention to Improve Unsupervised Learning. In: *From Animals To Animats, Proceedings of the Third International Conference on the Simulation of Adaptive Behavior*, Cliff, D., P. Husbands, J.A. Meyer, and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Gallistel, C.R. (1980). *The Organization of Action: A New Synthesis.* Lawrence Erlbaum Associates, Hillsdale N.J.

Gallistel, C.R. (1990). *The Organization of Learning.* MIT Press, Cambridge Ma.

Gallistel, C.R. (1994). Space and Time. In: *Animal Learning and Cognition.* Mackintosh, N.J. ed. Academic Press, San Diego.

Galyean, T. A.(1995). *Narrative Guidance of Interactivity.* Ph.D. Dissertation, Massachusetts Institute of Technology.

Girard, Michael and A. A. Maciejewski (1985). Computational Modeling for the Computer Animation of Legged Figures. In: *Proceedings of SIGGRAPH 85.*

Giszter, S. (1994). Reinforcement Tuning of Action Synthesis and Selection in a Virtual Frog. In: *From Animals To Animats, Proceedings of the Third International Conference on the Simulation of Adaptive Behavior*, Cliff, D., P. Husbands, J.A. Meyer, and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Gould, J.L. (1982). *Ethology: The Mechanisms and Evolution of Behavior.* W.W. Norton, New York

Gould, J.L. and C.G. Gould (1994). *The Animal Mind.* Scientific American Library, New York.

Griffin, D.R. (1992). *Animal Minds.* University of Chicago Press, Chicago.

Halperin, J. (1991). Machine Motivation. In: *From Animals To Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior*, Meyer, J.A. and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Hogan,J. (1988) Cause and function in the development of behavior systems. *Handbook of behavioral neurobiology, Vol 9.* Blass, E. ed. Plenum Press, New York.

Hodgdon, H. (1978). *Social Dynamics and Behavior within an Unexploited Beaver Population.*

Ph.D. Thesis, University of Massachusetts.

Holland, J. (1986). Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. In: *Machine Learning, an Artificial Intelligence Approach, Vol. 2*. R.S. Michalski R.S., J.G. Carbonell, and T.M. Mitchell eds. Morgan-Kaufmann, San Mateo

Holland, P.C. (1992). Event Representation in Pavlovian conditioning: Image and action. In: *Animal Cognition.* Gallistel, C.R. ed. MIT Press, Cambridge.

Horswill, I. (1993). A Simple, Cheap, and Robust Visual Navigation System. In: *Second International Conference on the Simulation of Adaptive Behavior.* Honolulu, HI. MIT Press

Humphrys, M.(1996). Action Selection Methods using Reinforcement Learning. In: *From Animals To Animats, Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior*, September 1996, MIT Press. Cambridge Ma.

Johnson, M. (1995). *WavesWorld: A Testbed for Three Dimensional Semi-Autonomous Animated Characters.* Ph.D. Thesis, Media Lab, MIT.

Jones, C. (1992). *Chuck Amuck: The life and times of an animated cartoonist*, Farrar, Straus Giroux. New York.

Kaelbling, L. (1992). *Learning in Embedded Systems*, MIT Press, Cambridge Ma.

Killeen, P.R. (1994). Mathematical principles of reinforcement. *Behavioral and Brain Sciences*, 17,105-172.

Klopf, A.H., J.S. Morgan, and S.E. Weaver (1993). A Hierarchical Network of Control Systems that Learn: Modeling Nervous System Function During Classical and Instrumental Conditioning. *Adaptive Behavior,* Vol. 1, No. 3.

Koga, Yoshihito, Koichi Kondo, James Kuffner, and Jean-Claude Latombe (1994). Planning Motion with Intentions. In: *Proceedings of SIGGRAPH 94.*

Krebs, J.R. and N.B. Davies (1993). *An Introduction to Behavioral Ecology.* Blackwell Scientific Publications, London UK.

Lasseter, J. (1987). Principles of Traditional Animation Applied to 3D Computer Animation. In: *Proceedings of SIGGRAPH 87.*

Latombe, J. C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Boston.

LeDoux, J. (1990). Information Flow from Sensation to Emotion: Plasticity in the Neural Computation of Stimulus Value. In: *Learning and Computational Neuroscience: Foundations of Adaptive Networks*. Gabriel, M. and J. Moore, eds. MIT Press, Cambridge Ma.

Lin, L.J. (1991). Self-improving reactive agents based on reinforcement learning, planning and teaching. In: *Machine Learning,* 8(3/4).

Lin, L.J., and T.M. Mitchell (1992). *Memory Approaches to Reinforcement Learning in Non-Markovian Domains,* CMU-CS-92-138, Dept. of Computer Science, Carnegie-Mellon University.

Lorenz, K. (1973). *Foundations of Ethology.* Springer-Verlag, New York.

Lorenz, K. (1994). *Man Meets Dog.* Kodansha America. New York.

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

Ludlow, A. (1980). The Evolution and Simulation of a Decision Maker. In: *Analysis of Motivational Processes,* Halliday F.T. &T. eds. Academic Press, London.

Maes, P. (1990a). Situated Agents Can Have Goals. *Journal of Robotics and Autonomous Systems* 6(1&2).

Maes P. & R. Brooks (1990b). Learning to Coordinate Behaviors. In: *Proceedings of AAAI-90*.

Maes, P. (1990). Learning Behavior Networks from Experience. In: *Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life.* Varela F.J. & P. Bourgine eds. MIT Press, Cambridge.

Maes, P. (1994). Modeling Adaptive Autonomous Agents. *Artificial Life*, Vol. 1, Numbers 1&2.

Maes, P., 1995, Artificial Life meets Entertainment: Interacting with Lifelike Autonomous Agents, In: *Communications of the ACM Special Issue on Novel Applications of A.I.*

Maes, P., T. Darrell, B. Blumberg, and A. Pentland (1996). The ALIVE System: Wireless, Full-Body Interaction with Autonomous Agents, In: The ACM Special Issue on Multimedia and Multisensory Virtual Worlds, spring 1996)

Mahadevan S. and J. Connell (1991). Automatic Programming of Behavior-Based Robots using Reinforcement Learning. In: *Proceedings of the Ninth National Conference on Artificial Intelligence*. MIT Press, Cambridge Ma.

Mataric M. (1990). Navigating with a rat brain: A neurologically-inspired model for robot spatial representation. In: *From Animals To Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior*, Meyer, J.A. and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Maudlin, M.(1994). *ChatterBots, TinyMuds and the Turing test: Entering the Loebner prize competition.* In: Proceedings of the AAAI 1994 Conference. MIT Press, Cambridge Ma.

McFarland, D. (1974). Time-Sharing as a Behavioral Phenomenon. In: *Advances in Animal Behavior, Vol. 5.* Academic Press, New York.

McFarland, D. (1976). Form and Function in the temporal organization of behavior. In: *Growing Points in Ethology*. Bateson P. & Hinde R. eds. Cambridge University Press.

McFarland, D. & R. Sibley (1975) The behavioral final common path. *Philosophical Transactions of the Royal Society*, B. 270.

McFarland, D. (1989). *Problems of Animal Behavior*. Longman Scientific and Technical, Harlow UK.

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

McFarland, D. and T. Bosser(1993). *Intelligent Behavior in Animals and Machines*. MIT Press., Cambridge Ma.

McKenna, Michael and D. Zeltzer (1990). Dynamic Simulation of Autonomous Legged Locomotion. In: *Proceedings of SIGGRAPH 90*

McCleery, R. (1983). Interactions between Activities. In: *Animal Behavior: Causes and Effects.* Vol. 1, Halliday, T. & Slater P. eds. W.H. Freeman & Co., New York.

McCallum, A.K.(1996). Learning to Use Selective Attention and Short-Term Memory in Sequential Tasks. In: *From Animals To Animats, Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior*, September 1996, MIT Press. Cambridge Ma

McNaughton B., C. Barnes, J. Gerrard et al (1996). Deciphering the Hippocampal Polyglot: The Hippocampus as a Path Integration System. *Journal of Experimental Biology.* 199, Jan. 1996.

Mel, B. (1995). Animal Behavior in Four Components. In: *Comparative Approaches to Cognitive Science.* Roitblat H.L. and J.A. Meyer, eds. MIT Press, Cambridge Ma.

Miller, G.F., and P.M. Todd (1990). Exploring adaptive agency I: Theory and methods for simulating the evolution of learning. In: *Proceedings of the 1990 Connectionist Models Summer School,* Touretzky D.S., J.L. Elman, T.J. Sejnowski, and G.E. Hinton, eds. Morgan Kaufmann, San Mateo, Ca.

Minsky, M. (1988). *The Society of Mind*. Simon & Schuster, New York.

Montague, P.R., P. Dayan, and T.J. Sejnowski (1994). Foraging in an uncertain environment using predictive Hebbian learning. In: *Advances in Neural Information Processing 6*, Cowan, J.D., Tesauro, G, and Alspector, J. eds. Morgan Kaufmann, San Mateo, Ca.

Mura, F. and N. Franceschini. (1994). Visual Control of Altitude and Speed in a Flying Agent. In: *From Animals To Animats, Proceedings of the Third International Conference on the Simulation of Adaptive Behavior*, Cliff, D., P. Husbands, J.A. Meyer, and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Pausch, R. and J. Snoddy, R. Taylor, S. Watson, E. Haseltine. (1996). Disney's Aladdin: First Steps Toward Storytelling in Virtual Reality. In: *Proceedings of SIGGRAPH 96.*

Payton, D.W. (1990). Internalized plans: A representation for action resources. In: *Designing Autonomous Agents*. P. Maes, ed. MIT Press, Cambridge.

Payton D., D. Keirsey, J. Krozel, And K. Rosenblatt, (1992). Do Whatever Works: A Robust Approach To Fault-tolerant Autonomous Control. In: *Journal Of Applied Intelligence, Vol. 3.*

Perlin, K. (1995). Real Time Responsive Animation with Personality. In: *IEEE Transactions on Visualization and Computer Graphics, March, 1995, Vol 1. Number 1.*

Perlin, K. and A. Goldberg. (1996). Improv: A System for Scripting Interactive Characters in Virtual Worlds. In: *Proceedings of SIGGRAPH 96.*

PF. Magic, (1996) *Dogz: Your Computer Pet.* Mac/IBM PC computer game

Plotkin, H.C. (1993). *Darwin Machines and the Nature of Knowledge*. Harvard University Press, Cambridge.

Porter, T. (1996) Personal communication.

Raibert, M. H. and J. K. Hodgins (1991). Animation of Dynamic Legged Locomotion. In: *Proceedings of SIGGRAPH 91*.

Ramachandran V.S. and S. M. Anstis (1990). The Perception of Apparent Motion. In: *The Perceptual World*. I. Rock, ed. W.H. Freeman, New York.

Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. In: *Proceedings of SIGGRAPH 87.*

Reynolds, G.S. (1975) *A Primer of Operant Conditioning.* Scott, Foresman and Company. Glenview, Illinois.

Renault, O., and N. Magnenat-Thalmann, D. Thalmann (1990). A vision-based approach to behavioral animation. In: *The Journal of Visualization and Computer Animation* 1(1).

Rhodes, B. (1995). Pronomes in Behavior Nets, Technical Report # 95-01, MIT Media Lab, Learning and Common Sense Section.

Rhodes, B. (1996). *PHISH Nets: Planning Heuristically in Situated Hybrid Networks*. Masters Thesis, The Media Lab, MIT.

Rogerson, J. (1992). *Training Your Dog.* Howell Book House, London.

Rosenblatt K., and D. Payton (1989). A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control. In: *Proceedings of the International Joint Conference on Neural Networks*, June 1989, Washington DC.

Russell, K., B. Blumberg, A. Pentland, P Maes (1996) *Distributed Alive*. In: Technical Sketches, Visual Proceedings of Siggraph 96.

Schmajuk, N.A. and H.T. Blair (1993). Place Learning and the Dynamics of Spatial Navigation: A Neural Network Approach. In: *Journal of Adaptive Behavior*, Vol. 1, No. 3.

Schmajuk, N.A. (1994).Behavioral Dynamics of Escape and Avoidance: A Neural Network Approach. In: *From Animals To Animats, Proceedings of the Third International Conference on the Simulation of Adaptive Behavior*, Cliff, D., P. Husbands, J.A. Meyer, and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Shanks, D.R. (1994). Human Associative Learning. In: *Animal Learning and Cognition,* Mackintosh, N.J. ed. Academic Press, San Diego.

Shettleworth, S.J. (1994). Biological Approaches to the Study of Learning. In: *Animal Learning and Cognition,* Mackintosh, N.J. ed. Academic Press, San Diego.

Sims, K. (1994). Evolving Virtual Creatures. In: *Proceedings of SIGGRAPH 94*.

Srinvasan, M., S. Zhang, M. Lehrer, and T. Collett (1996). Honeybee Navigation en route to the Goal: Visual Flight Control and Odometry. *Journal of Experimental Biology.* 199, Jan. 1996.

Sutton, R., and A.G. Barto (1990). Time-Derivative Models of Pavlovian Reinforcement. In: *Learning and Computational Neuroscience: Foundations of Adaptive Networks.* Gabriel, M. and J. Moore, eds. MIT Press, Cambridge Ma.

Sutton R. (1991). Reinforcement Learning Architectures For Animats. In: *From Animals To Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior*, Meyer, J.A. and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Terzopoulos, D. and T. Rabie, R. Grzeszczuk (1996). Perception and Learning in Artificial Animals. In: *Artificial Life V: Proceedings of the Fifth International Conference on the Synthesis and Simulation of Living Systems*, Nara Japan.

Thomas, F. and O. Johnson (1981). *The Illusion of Life, Disney Animation.* Hyperion, New York.

Tinbergen, N. (1951). *The Study of Instinct.* Oxford University Press, New York.

Timberlake, W. and G. Lucas (1989). Behavior Systems and Learning: From Misbehavior to General Principles. In: *Contemporary Learning Theories: Instrumental Conditioning Theory and the Impact of Biological Constraints*. Klein S. & Mowrer, R., eds, Lawrence Erlbaum Associates, Inc. Hillsdale NJ.

Toates, F. (1983). Models of Motivation. In: *Animal Behavior: Causes and Effects, Vol. 1*. Halliday, T. & Slater P. eds. W.H. Freeman & Co., New York.

Toates F. (1985). Animal Motivation and Cognition. In: *Comparative Approaches to Cognitive Science.* Roitblat H.L. and J.A. Meyer, eds. MIT Press, Cambridge Ma.

Toates F. and P. Jensen (1991). Ethological and Psychological Models of Motivation: Towards a Synthesis. In: *From Animals To Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior.* Meyer, J. and Wilson, S.W. eds. MIT Press, Cambridge Ma.

Todd, P.M. and G.F. Miller (1991). Exploring adaptive agency II: Simulating the evolution of associative learning. In: *From animals to animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, Meyer J.A., and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Todd, P.M., and G.F. Miller (1991). Exploring adaptive agency III: Simulating the evolution of habituation and sensitization. In: *Proceedings of the First International Conference on Parallel Problem Solving from Nature*. Schwefel, H.P. and R. Maenner eds. Springer-Verlag, Berlin.

Touretzky, D. and L. Saksida(1996). Skinnerbots. To be published in: *From Animals To Animats, Proceedings of the Fourth International Conference on the Simulation of Adaptive Behavior*, September 1996, MIT Press. Cambridge Ma.

Travers, M. (1988). *Agar: An Animal Construction Kit*. Ph.D. Thesis, The Media Lab, MIT.

Tu, Xiaoyuan and D. Terzopoulos (1994). Artificial Fishes: Physics, Locomotion, Perception, Behavior. In: *Proceedings of SIGGRAPH 94*.

Tyrrell T. (1993). *Computational Mechanisms for Action Selection.* Ph.D. Thesis, Centre for Cognitive Science, University of Edinburgh.

Vershure, P.F., J. Krose, and R. Pfeifer (1991). Distributed Adaptive Control: The self-organization of structured behavior. *In: Robotics and Autonomous Systems*, 9.

Watkins C. (1989). *Learning from Delayed Rewards*. Ph.D. Thesis, King's College, Cambridge.

Wehner R., B. Michel and P. Antonsen (1996). Visual Navigation in Insects: Coupling of Egocentric and Geocentric Information. *Journal of Experimental Biology.* 199, Jan. 1996.

Whitehead S.D. (1992). *Reinforcement Learning for the Adaptive Control of Perception and Action*. Technical Report 406, University of Rochester Computer Science Dept.

Wilhelms J. and R. Skinner (1990). A 'Notion' for Interactive Behavioral Animation Control. *IEEE Computer Graphics and Applications 10(3)* May 1990.

Wilson, S.E. (1991). The Animat Path to AI. In: *From Animals To Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior*, Meyer, J.A. and S.W. Wilson, eds. MIT Press, Cambridge Ma.

Wren, C., T. Darrell, T. Starner, M. Johnson, K. Russell, A. Azerbanjani, and A. Pentland (1995). *pfinder: A Real-Time System for Tracking People.* In: Proceeedings of the SPIE Conference on Real-Time Vision, Bove, M. ed., Philadelphia Pa.

Zeltzer, D. (1991). Task Level Graphical Simulation: Abstraction, Representation and Control. In: *Making Them Move*. Badler, N., Barsky, B., and Zeltzer D. eds. Morgan Kaufmann Publishers, Inc., Menlo Park, Ca.

10.0 Appendix

Here we list the Degrees of Freedom, Motor Skills and Motor commands which comprise Silas.

10.1 Degrees of Freedom

Right_front_legControls leg (3 joints) via foot position (i.k.)
Left_front_legControls leg (3 joints) via foot position (i.k.)
Left_back_legControls leg (3 joints) via foot position (i.k.)
Right_back_legControls leg (3 joints) via foot position (i.k.)
Shoulder_y_jointRotates shoulder around y axis
Shoulder_jointRotates shoulder around z axis
Shoulder2Back_y_jointRotates back around y axis of shoulder
Shoulder2Back_z_jointRotates back around z axis of shoulder
Back2Hip_y_jointRotates hip around y axis of back
Back2Hip_z_jointRotates hip around z axis of back
Tail_y_jointControls wagging tail
Tail_z_jointControls height of tail
Neck_jointControls height of neck
Head_jointControls head and neck via gaze point (i.k.)
Head_tilt_jointRotates head around neck
Left_eyeball_jointRotates eyeball relative to head
Right_eyeball_jointRotates eyeball relative to head
Pupil_jointDilates/Contracts pupils
Eyebrow_updown_jointRaise/lower eyebrows relative to eyeballs
Left_eyebrow_jointRotates eyebrow relative to eyeball
Right_eyebrow_jointRotates eyebrow relative to eyeball
Left_ear_y_jointMoves ear forward/back
Right_ear_y_jointMoves ear forward/back
Ear_tip_jointControls bend in ears
Jaw_jointOpen/closes mouth
Tongue_jointExtends/Retracts tongue
Nose_jointControls “sniffing”
Show_teeth_jointControls visibility of teeth

10.2 Motor Skills

- Silas Specific

  Walking

Turning
WaggingAndMovingTail
Sitting
Begging
LiftLeg
ShakePaw
Crouch
LyingDown
Sniffing
RotatingEars
MovingEyeBrows
ShowTeeth
BugEyes
LookAt
LookAt
ShakeHead
NodHead
Bark
OpeningAndClosingMouth
NoddingAndLoweringHead

- Inherited Motor Skills from Motor Creature

GenericMove
GenericTurn
ListCommands
SetTransformSkill
PositionSkill
getPositionSkill
getFloorPositionSkill
getHeadingSkill
getFloorHeadingSkill
HeadingSkill
HeadTowardSkill
GrabberSkill
PositionRelSkill
HeadingRelSkill
SoundSkill
WriteBehaviorFile
ResetAction

ChangeRep
SetVecField
SetBoolFieldGeneric
SetStringFieldGeneric
SetFloatFieldGeneric
SetVecFieldInNamedNode
SetBoolFieldInNamedNode
SetFloatFieldInNamedNode
GetVecField
GetBoolFieldGeneric
GetStringFieldGeneric
GetFloatFieldGeneric
GetVecFieldInNamedNode
GetBoolFieldInNamedNode
GetFloatFieldInNamedNode
GetBoolFields

- **Inherited Motor Skills from Behavior Creature**

SetRenderQuality
SetCameraPosition
GetCameraPosition
SetCameraOrientation
GetCameraOrientation
SetBehaviorVariable
SetBoolField
GetBehaviorStateVector
GetShortTermMemory
GetMotStateVector
GetRMStateVector
GetPerceptFields
GetPerceptFieldValues
BuildBooleanRMs
BuildDiscoveryGroup
AddAppetitiveBehavior
GetTopBehaviorWithIV
UpdateChangeDetector

10.3 Major Classes Provided by Hamsterdam

Below we list a number of the important classes from which a behavior system is constructed:

- **Behavior Group**: This subclass of SoGroup has one or more children which are of type Behavior. During action-selection all of the behaviors within a Behavior Group compete via the mutual inhibition mechanism described earlier.
- **Behavior**: This subclass of SoGroup implements the basic functionality associated with behaviors. It typically has at least one child of type DependsOnVariables which is used to hold the Releasing Mechanisms and InternalVariables upon which the Behavior depends for its value (see below). It may have zero or more children of type ModifyVariable, if it is to affect the value of an Internal Variable when it is active. Finally, it will have a child of type Behavior Group to hold any children behaviors which are associated with it. Behaviors have two associated callbacks. One callback is invoked when the behavior is active, and it is a leaf behavior, and this callback is responsible for issuing the appropriate motor commands for when the behavior is active. The other callback is invoked when the Behavior is in a Behavior Group which is traversed during action-selection, but the Behavior itself is a loser in the arbitration process. This callback is used to issue any suggestions that the Behavior might have (i.e. secondary or meta-commands).
- **DependsOnVariables**: This subclass of SoGroup acts as a container to hold all of the Releasing Mechanisms and InternalVariables associated with a given Behavior. During action-selection it is responsible for having each of its children (i.e. the Releasing Mechanisms and InternalVariables) update their value, and then it aggregates their respective values using one of a number of functions (e.g. addition or multiplication).
- **ModifyVariable**: This subclass of SoNode is used to specify an InternalVariable whose value is to be adjusted when the Behavior is active, and a gain to be used. The gain may be absolute (i.e. the InternalVariable is adjusted by the amount of the gain), or used as a multiplicative factor in conjunction with the associated Behavior (i.e. the InternalVariable is adjusted by an amount equal to the product of the gain and the value of the associated Behavior). This adjustment happens once per tick.
- **InternalVariable**: This subclass of SoNode is used to implement time-varying motivational variables as described earlier. During action-selection it calculates and returns its current value. It typically has a per tick growth rate and damping rate associated with it, as well maximum and minimum limits for its allowed value.
- **Releasing Mechanism**: This subclass of SoGroup is used to implement Releasing Mechanisms as described earlier. During action-selection it updates its value based on the presence or lack thereof of relevant stimuli. It has a number of parameters which specify things like the kind of object to which it is sensitive, or the minimum, maximum and optimal distance for the Releasing Mechanism's object of interest. The optimal distance is the distance at which the Releasing Mechanism will reach its allowed maximum value (also specified via a parameter). In addition, to these parameters, Releasing Mechanisms can have up to three different callbacks associated with them: a find callback used to find the RM's object of interest, a filter callback which is used to filter the attributes of the object of interest for more specific criteria (e.g. that the hand is low and extended), and a weight callback which is

responsible for weighting the value of the RM based on some criteria such as distance (i.e. the callback calculates a value between 0 and 1 which is applied against the maximum allowed value for the RM). Releasing Mechanisms may also do some type of temporal filtering such as averaging or summing over a period or implementing a kind of latch in which the RM is active if the relevant stimuli was detected at anytime within a given period. For flexibility Releasing Mechanisms may have ModifyVariables as children if it is desired that they affect an InternalVariable directly when they are active.

There are a number of subclasses of Releasing Mechanism, most notably CreatureFieldRM which has built-in support for using relevant boolean fields in the RM's object of interest as predicates. Specifically, CreatureFieldRM has a field which contains one or more strings which refer to the names of the fields in the object of interest to be used as predicates, and another field which specifies the logical operator to use (AND,OR, or NAND).

In addition to these major classes, there are a number of subsidiary classes including the subclasses of SoAction which together with the classes described above implement the action selection and learning algorithms.