

78

Imitation and Social Learning For Synthetic Characters

by

Daphna Buchsbaum

A.B., Brown University (2002)

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

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

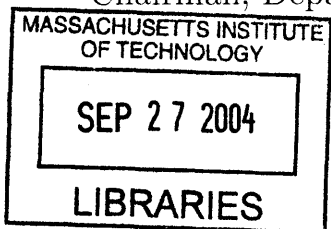
September 2004

© Massachusetts Institute of Technology 2004. All rights reserved.

Author
Program in Media Arts and Sciences,
School of Architecture and Planning
August 6, 2004

Certified by
Bruce M. Blumberg
Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by
Andrew Lippman
Chairman, Departmental Committee on Graduate Students
Program in Media Arts and Sciences



ROTC

Imitation and Social Learning For Synthetic Characters

by

Daphna Buchsbaum

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on August 6, 2004, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

We want to build animated characters and robots capable of rich social interactions with humans and each other, and who are able to learn by observing those around them. An increasing amount of evidence suggests that, in human infants, the ability to learn by watching others, and in particular, the ability to imitate, could be crucial precursors to the development of appropriate social behavior, and ultimately the ability to reason about the thoughts, intents, beliefs, and desires of others.

We have created a number of imitative characters and robots, the latest of which is Max T. Mouse, an anthropomorphic animated mouse character who is able to observe the actions he sees his friend Morris Mouse performing, and compare them to the actions he knows how to perform himself. This matching process allows Max to accurately imitate Morris's gestures and actions, even when provided with limited synthetic visual input. Furthermore, by using his own perception, motor, and action systems as models for the behavioral and perceptual capabilities of others (a process known as Simulation Theory in the cognitive literature), Max can begin to identify simple goals and motivations for Morris's behavior, an important step towards developing characters with a full theory of mind. Finally, Max can learn about unfamiliar objects in his environment, such as food and toys, by observing and correctly interpreting Morris's interactions with these objects, demonstrating his ability to take advantage of socially acquired information.


Thesis Supervisor: Bruce M. Blumberg
Title: Associate Professor of Media Arts and Sciences

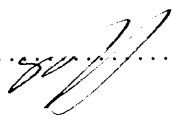
Imitation and Social Learning for Synthetic Characters

by

Daphna Buchsbaum

The following people served as readers for this thesis:

Reader

Cynthia Breazeal
Assistant Professor of Media Arts and Sciences
Program in Media Arts and Sciences
Massachusetts Institute of Technology

Reader

Andrew Meltzoff
Professor of Psychology
University of Washington

Acknowledgments

Good advice is always certain to be ignored, but that's no reason not to give it. -Agatha Christie

I've been fortunate to receive a lot of good advice during my time at the Media Lab, some of which I've even been sensible enough to listen to. First and foremost in providing me with thoughtful guidance has been my advisor Bruce Blumberg, who not only introduced me to the world of autonomous animated creatures, but even let himself be talked into using a character that isn't canine. I'd also like to thank my readers, Cynthia Breazeal and Andrew Meltzoff, for their thoughtful commentary on this manuscript, and helpful advice on the implementation of my thesis.

A number of other people have been instrumental to this work. I am an acknowledged administrative nightmare, and so I'd like to thank Aileen Kawabe, Linda Peterson and Pat Solakoff for helping me keep my act together and making sure I graduate. I'd also like to thank the other members of the Synthetic Characters Group: Matt Berlin for all his help easing the thesis process, Jesse Gray (an honorary character) for making things go, Derek Lyons for going through the growing pains of a Masters degree with me, and Jennie Cochran for improving the group's male:female ratio. Special thanks to Marc Downie, who I can never repay in advice and expertise, so it will have to be in blood. Past group members Matt Grimes, Lily Shirvane, Marta Luczynska and Seth Block all contributed to an exciting environment. Chris Leathers is a fabulous animator and I'd like to thank him for his animation support. Ari Benbasat and Josh Lifton, while not characters, have been remarkably effective at distracting me from my work. Dan Stiehl, a former character, knows where I live.

I'd also like to thank my parents, Maya and Gershon Buchsbaum, and my grandmother, Irena Hecht, who have been advising me for almost 25 years (and counting), my sisters Nilly and Talia, two of the best people I've ever met and the Tel Aviv branch of my family, who have provided moral support from afar. The Morses, Doug and Elsie, have been an incredible source of advice and homemade bread. Dan Goldwater isn't one for sentiment. He is, however, a monkey. And the former residents of 46/48 East George St. are perhaps the luckiest thing that's ever happened to me—thanks for everything.

Contents

1	Introduction	19
1.1	Motivation	19
1.1.1	Socially Intelligent Characters and Robots	19
1.1.2	Natural Systems	20
1.2	Approach	21
1.3	Contributions	21
1.3.1	What This Thesis Does	21
1.3.2	What This Thesis Doesn't Do	22
1.4	Roadmap	23
2	Lessons from Social Learning in Humans and Animals	25
2.1	Imitation and Social Learning in Animals	26
2.1.1	Imitation and Hierarchical Action Structures	28
2.2	Learning About Others	30
2.2.1	Understanding Other's Minds	31
2.2.2	Simulation Theory	32
2.2.3	Imitation and Simulation Theory	32
2.2.4	Mirror Neurons	33
2.2.5	Understanding Observed Actions	34
2.2.6	From Social Animals to Social Characters	37
3	Background and Related Work	39
3.1	Social characters and robots	40

3.1.1	Interactive Animated Characters	40
3.1.2	Social Robots	42
3.2	Imitative characters and robots	44
3.2.1	Learning new actions	44
3.2.2	Learning to imitate	46
3.2.3	Learning by imitation	48
3.2.4	Towards Socially Intelligent Characters and Robots	49
4	Max and Morris	51
4.1	Sample Interaction :The Basics	52
4.2	Sample Interaction: Imitation	54
4.3	Sample Interaction: Action Identification	55
4.4	Sample Interaction: Learning About Objects I	57
4.5	Sample Interaction: Learning About Objects II	57
4.6	Looking Ahead	58
5	Cognitive Architecture	61
5.1	System Overview	61
5.2	Input to the System	62
5.3	Sensory System	63
5.3.1	Synthetic Vision	64
5.4	Perception System	66
5.5	Belief System	68
5.5.1	Belief Selectors	69
5.5.2	Derived Percepts	69
5.6	Action System	70
5.6.1	Motivations, Drives and Autonomic Variables	72
5.7	Motor System	74
5.7.1	The Posegraph	75
5.7.2	Moving around the graph	75
5.7.3	Multi-resolution graphs	76

5.7.4	Motor Programs	77
5.8	Summary	79
6	Implementation and Results	81
6.1	Imitation and Movement Recognition	81
6.1.1	Overview	81
6.1.2	Parsing Observed Motion into Gestures	82
6.1.3	Matching Observed Gestures to Movements in the Graph	88
6.1.4	Imitation	97
6.2	Identifying Actions, Motivations and Goals	98
6.2.1	Action Identification: Example 1	99
6.2.2	Action Identification: Example 2	103
6.2.3	Step-By-Step Summary of Action Identification	108
6.2.4	Representing Self and Other	109
6.2.5	Motivations and Goals	110
6.2.6	Learning About Objects	111
6.3	Results	113
7	Discussion and Future Work	117
7.1	Stumbling Blocks, Successes and Surprises	117
7.1.1	Simulation Theory as the Road to Social Characters	118
7.1.2	Limits of Simulation Theory	122
7.2	Future Work	124
7.2.1	Solving the Correspondence Problem	125
7.2.2	Other Problems in Imitating Humans	126
7.2.3	Understanding Emotions	127
7.2.4	Cooperative Behavior	128
7.2.5	Predicting Future Actions	129
7.2.6	Learning New Movements	130
7.2.7	Learning New Actions	130
7.2.8	Learning New Goals	131

A Synthetic Vision	133
B Terminology	135
Bibliography	137

List of Figures

2-1	An example motivational system for animal feeding (after Timberlake 1989 [110])	29
2-2	This is a hypothetical example, where primate Y watches primate X. By positing an internal state of ‘wanting’ in primate X, Y can gain economy of representation (from Whiten 1996 [120]).	37
3-1	two wolf pups in the Alphawolf installation interacting.	41
3-2	Kismet.	42
3-3	The robot Ripley (image taken from www.media.mit.edu/cogmac) . .	43
3-4	An animated version of the Leonardo robot imitates a human model’s facial expressions.	50
4-1	Max (right) and Morris (left) in the virtual desert	51
4-2	The objects that can be introduced into the world.	52
4-3	Visualizer showing the current level of Max’s hunger and play drives .	52
4-4	Buttons for introducing objects into the world	53
4-5	Max (left) and Morris (right) jumping for a piece of cheese	53
4-6	Morris, rendered from Max’s point of view. Certain key effectors, such as Morris’s hands, nose and feet, are marked by colored sphere’s . . .	54
4-7	Buttons for requesting that Morris perform different movements . . .	55
4-8	Buttons for requesting that Max observe, imitate or identify Morris’s action	55
4-9	Morris covering his eyes, as seen by Max. Notice that some of the spheres marking his body parts are not visible (compare to figure 4-6)	56

4-10	First row. Morris (blue) demonstrates an action (covering his eyes) while Max (brown) watches. Second row: Morris through Max's eyes. The colored spheres represent key effectors. Third row. Max reproduces Morris's action, by performing the movements in his own repertoire that are closest to what he observed.	59
5-1	This is the overall cognitive architecture used for characters such as Max and Morris. The systems are processed serially, in roughly top-to-bottom (or light-to-dark) order, relative to this figure. Imitation, and other social skills, make particular use of the perception, action and motor systems.(Adapted from Burke 2001 [34])	62
5-2	An overview of the Synthetic Character's System (After Isla 2001 [65])	63
5-3	This figure shows Morris in 3 poses. The top row is Morris as we see him, while the bottom row is Morris as seen through Max's synthetic vision. The colored spheres on Morris's body are key body parts whose location is tracked by the synthetic vision system.	65
5-4	A simple example percept tree.	66
5-5	An example action system. Purple rectangles represent tuples. Red circles are trigger contexts, yellow triangles are objects, and blue rectangles are actions (do-until contexts not shown). There are three motivational subsystems in this example action system.	70
5-6	A simplified diagram of Max's <i>hunger</i> motivational subsystem (the top-level of his <i>play</i> motivational subsystem is shown as well)	72
5-7	An autonomic variable, the atomic component of internal representation (from Burke 2001 [33])	73
5-8	A simple posegraph. Green lines are allowable transitions between poses. The Blue square is the pose representing the characters current body configuration.	75

5-9	An example graph of movement nodes. Large rectangles represent movements, small squares represent poses. Stacks represent blended movements and poses	77
6-1	An example graph of movement nodes. Large rectangles represent movements, small squares represent poses. Stacks represent blended movements and poses (this figure is the same as figure 5-9)	83
6-2	Morris viewed from Max's perspective. The colored sphere marking his root node is circled in yellow.	84
6-3	This figure shows 4 frames of Morris moving into a jump. In frame 1 Morris starts in a standing position. by frame 3 his body position has changed enough to cross the threshold between standing and moving.	86
6-4	The movement matching process—step 1	90
6-5	the movement matching process—step 2	90
6-6	The movement matching process—step 3	91
6-7	The movement matching process—step 4	91
6-8	The movement matching process—step 5	92
6-9	The movement matching process—step 6	92
6-10	The movement matching process—step 7	93
6-11	The movement matching process—step 8	93
6-12	The movement matching process—step 9	93
6-13	The movement matching process—step 10	94
6-14	Morris covering his eyes, as seen by Max. Notice that some of the spheres marking his body parts are not visible. This is a repeat of figure 4-9.	97
6-15	An example action system. Purple rectangles represent tuples. Red circles are trigger contexts, yellow triangles are objects, and blue rectangles are actions (do-until contexts not shown). This figure is a repeat of figure 5-5.	98

6-16	A simplified diagram of Max's <i>hunger</i> motivational subsystem (the top-level of his <i>play</i> motivational subsystem is shown as well). This figure is the same as figure 6-16 seen earlier.	99
6-17	Identifying where the <i>eat</i> movement is used	100
6-18	the path through Max's action system to his eating action	101
6-19	Evaluating the <i>can-I triggers</i> along the path to Max's eating action	102
6-20	The paths through Max's action hierarchy to the <i>reach</i> action. Notice that it can be activated in both the <i>hunger</i> and <i>play</i> motivational subsystems	104
6-21	When Max reaches for a piece of cheese in order to satisfy his hunger, the <i>satisfy hunger</i> , <i>get</i> and <i>reach</i> action tuples are marked as having been active during the <i>reaching</i> movement in Max's movement-action correspondence map	105
6-22	Max evaluates the <i>can-I triggers</i> along the path from the <i>play</i> tuple to the <i>reach</i> tuple	106
6-23	Max evaluates the <i>can-I triggers</i> along the path from the <i>satisfy hunger</i> tuple to the <i>reach</i> tuple	107
6-24	The <i>play</i> motivational subsystem of the action hierarchy. Notice that the <i>dance</i> and <i>throw</i> tuples have more than one <i>can-I trigger</i> —a <i>proximity trigger (holding object)</i> which checks whether the <i>object of attention</i> is in Max's hands, and an <i>object selection trigger (Danceable object and Throwable object)</i> , which checks whether the <i>object of attention</i> is of the appropriate type for that tuple to act on.	112
6-25	Top row: Morris demonstrates jumping. Bottom row: Max imitates jumping.	115
6-26	Top row: Morris demonstrates pounding the ground. Bottom row: Max imitates pounding the ground.	115
6-27	Top row: Morris demonstrates giving a thumbs up. Bottom row: Max imitates giving a thumbs up.	116

6-28	Left: Morris reaches for the piece of cheese. Right: Max identifies Morris's action as reaching for the cheese.	116
6-29	Left: Morris eats the piece of cheese. Right: Max identifies Morris's action as eating.	116

Chapter 1

Introduction

Humans, and many other animals, display a remarkably flexible and rich array of social competencies, demonstrating the ability to interpret, predict and react appropriately to the behavior of others, and to engage others in a variety of complex social interactions. Developing systems that have these same sorts of social abilities is a critical step in designing robots, animated characters, and other computer agents, who appear intelligent and capable in their interactions with humans and each other, who are intuitive and engaging for humans to interact with, and who maximize their ability to learn from the world around them. The aim of this thesis is to work towards the ultimate goal of socially intelligent artificial systems, by creating animated characters capable of interpreting and learning from the actions and intentions of others.

1.1 Motivation

1.1.1 Socially Intelligent Characters and Robots

Some of the most exciting new applications being developed for synthetic creatures require them to cooperate with humans and each other as socially capable partners. For instance, animated characters and robots are being developed as individualized tutors for children. Such a character should be encouraging and persuasive in ways

that are sensitive to the child, adjusting to their learning style and current mood, in order to hold or reclaim attention. In general, characters and robots that interact with people need to respond with social appropriateness, and they must be easy for the average person to use and relate to. They must also be able to quickly learn new skills and how to perform new tasks from human instruction and demonstration. Ideally, programming such a character with new capabilities would be as easy as showing it what to do. Finally, to cooperate with humans as capable partners artificial creatures must be able to interpret our behaviors and emotions, so that they can provide us with well-timed, relevant assistance.

1.1.2 Natural Systems

As character designers, it is possible to gain valuable insights into how social intelligence might operate and be acquired by looking to the fields of developmental psychology and animal behavior. It appears that, among animals, learning from the behavior of others (known as social learning) is by no means a single monolithic process. Rather, species sample widely from a spectrum of overlapping social competencies [122], ranging from using information about others to help focus their attention, to emulating other's actions and goal states.

While very few species exhibit the most complex forms of imitation, and perhaps no non-human animal possesses a full theory of mind [93], the abilities animals do possess allow them to consistently exploit their social environment in ways that far outstrip our current technologies. Furthermore, many of the simpler behavior-reading abilities present in animals may represent prerequisites for the more complex mind-reading abilities humans possess. An increasing amount of evidence suggests that, in human infants, the ability to learn by watching others, and in particular, the ability to imitate, could be crucial precursors to the development of appropriate social behavior, and ultimately the ability to reason about the behaviors, emotions, beliefs, and intents of others [86] [84] [87].

1.2 Approach

In previous work, we began to explore the role of imitation and social learning in artificial intelligence, by implementing a facial imitation architecture for an interactive humanoid robot [28]. In this thesis, I present a novel system that provides artificial creatures with a cognitive architecture inspired by the literature on animal social learning, including a robust mechanism for observing and imitating whole gestures and movements. Critically, the characters presented in this thesis are able to use their imitative abilities to bootstrap simple mechanisms for identifying each other's low-level goals and motivations and learning from each other's actions, bringing us several steps closer to the goal of creating socially intelligent artificial creatures.

1.3 Contributions

1.3.1 What This Thesis Does

This work concerns the creation of synthetic creatures capable of a number of interesting and novel forms of social learning, inspired by the cognitive literature. In particular, characters with the following capabilities were implemented for this thesis:

- Correctly imitating and identifying observed gestures and movements after a single demonstration. Furthermore, the characters in this thesis observe each other using synthetic vision, and imitate each other using purely visual data.
- Identifying higher-level goal-directed behaviors, such as reaching for an object.
- Identifying potential motivations and goals for another character's actions, such as a desire to satisfy hunger, or to possess an object.
- Learning based on observing other character's behavior, such as learning about a new food object by watching another character consume it.

The goal of this thesis was not only to create synthetic creatures capable of learning by observing each other, but to test out theories from the cognitive literature while

doing so. To this end, this work hopes to make two additional contributions:

- Testing the prominent theory that imitative abilities help bootstrap social learning skills in humans, and the related idea that Simulation Theory (described in section 2.2.2) can be used to understand other’s actions, motivations and goals.
- Discovering underlying similarities or shared mechanisms among the large variety of social learning abilities hypothesized in the cognitive literature.

1.3.2 What This Thesis Doesn’t Do

Social learning and intelligence in artificial systems represents a vast research area, and this thesis is necessarily limited to a sub-section of the potential topics that fall under this rubric. The following is a list of topics and approaches outside the scope of this thesis (though a number of them have been anticipated as application areas for this work):

- While the work in this thesis is inspired by theories in the cognitive and animal behavior literatures, it is not meant to implement the details of any specific model of how animals and humans learn from each other. Similarly, ideas from the cognitive literature are implemented within this thesis at a purely representational level—this thesis does not address the neural substrate underlying these abilities in humans and animals.
- The system presented in this thesis was designed to be general enough for use with both animated characters and robots. However, so far, it has only been tested using animated characters, and this is the application area that will be focused on in this work.
- Similarly, while I believe the approach described in this work may be generalizable to human-robot and human-character interactions, this thesis uses character-character interactions as its starting point.
- The primary focus of this work is on imitation as a means of achieving other social learning capabilities, rather than imitation as an end-goal in itself. As

a result, this thesis does not delve deeply into problem areas such as imitating characters with different morphologies, or learning novel movement primitives through imitation.

- This work assumes that characters have very similar morphology, abilities and motivations. An expectation of sameness is actually a fundamental assumption of Simulation Theory, one of the primary theories of human cognition motivating this work (described in section 2.2.2).

1.4 Roadmap

To begin with, **Appendix B** of this thesis contains definitions of potentially ambiguous terminology used throughout this work, and may be worth consulting before venturing in too deeply. In the following chapter (**chapter 2**), I explore the cognitive theories motivating my approach to artificial social learning in a bit more detail. **Chapter 3** places this research in the context of previous work in interactive character design and social robotics. Subsequently, in **chapter 4** I introduce Max and Morris Mouse, two anthropomorphic animated mouse characters who are able to interact with each other, and observe each others' behavior. I then present a series of progressively more sophisticated results, in which Max the Mouse is initially able to imitate Morris, and is ultimately able to identify Morris's action-structure, including simple motivations and goals, and learn from Morris's actions. **Chapter 5** presents the details of the Synthetic Characters creature architecture used to create Max and Morris, while **chapter 6** explores the implementation of the social learning mechanisms at the heart of this thesis.

Finally, in **chapter 7** I discuss possible future work and extensions, and the implications of my results for both artificially intelligent agents and natural systems. It is worth noting that while the ultimate goal of this work is the development of socially intelligent artificial creatures, the approach presented here has the potential to contribute to a number of other research areas such as movement and gesture recognition, and motor system design for animated characters.

Chapter 2

Lessons from Social Learning in Humans and Animals

The natural world is teeming with examples of social behavior, from fish forming schools and birds forming flocks, to wolf pups tussling to determine dominance, to domestic cats begging their owners for food. In all of these cases, animals are able to interact and communicate appropriately with conspecifics in their environment (if you accept, for the moment, that you're a conspecific to your cat), in order to satisfy their individual motivations and goals (e.g. safety from predators, establishing place in pack hierarchy, acquiring food).

Within the broad range of social behaviors animals display, we are particularly interested in social learning, where “acquisition of behavior by one animal can be influenced by social interaction with others of its species” (Heyes and Galef 1996 [62], p.8). The ethological literature is filled with examples of animals learning by observing and interacting with others. Some classic instances include the spread of milk-bottle opening behavior among great tits in Britain [49], macaques learning to wash potatoes [64], and black rats learning how to pull the scales off of pine cones [108]. Many other well-studied behavior patterns can also be considered forms of social learning, such as juvenile song-birds learning species-specific song patterns by listening to adults [106], and alarm-call learning by young ground squirrels [79] and vervet monkeys [105].

There is a wide range of ways in which animals are able to learn from the presence of others, which run the gamut from simply interacting with objects others have left behind (as in the case of the black rats, who learn to shell pine cones by coming across the partially shelled remains left by adult rats), to learning an emotional stance towards an object by watching others interact with it [38] [40], to imitating either the results or the end-goal of another’s actions [114][83], to imitating the physical behaviors performed by another [84] [119]. Of these, the behaviors that traditionally fall under the auspices of social learning are those where one animal learns **directly** from the observed behavior of another (e.g. a chimpanzee using a rake to bring food closer, after seeing another chimpanzee perform the task), rather than those where they are indirectly influenced by another’s actions (e.g. the black rats). In the following sections, I will explore this kind of social learning, as well as its relationship to more general social intelligence, and the development of theory of mind.

2.1 Imitation and Social Learning in Animals

There is a rich research literature available investigating social learning in a variety of animal species, particularly non-human primates. Much of this literature has been devoted to partitioning socially mediated learning into various subtypes (for a review see [122], [36], [35] and also [62]). The primary contribution of this research to the design of socially adept artificial systems may lie not in the divisions between types of social learning that have occupied much of the research agenda, but rather in the spectrum of potential social learning situations and mechanisms these divisions highlight. Here, I draw attention to some of the most commonly cited ways in which one organism could potentially learn by observing another (the categories I use have been roughly adapted from Whiten [122]).

Attention Shifting. The animal’s attentional focus is affected by others actions.

This includes *stimulus enhancement*, where the observer becomes more likely to attend to and interact with stimuli it has noticed a model attending to.

Significance Learning Using other's behavior and reactions as cues about the significance of objects in the environment. This includes *social referencing*, where the observer alters his reaction to a stimulus based on the observed behavior of a model, and *affordance learning*, where the animal learns certain properties of the environment, or of objects in the environment, through observation.

Impersonation Copying the form of another's action. This category encompasses relatively simple behaviors such as *response facilitation*, where the observer becomes more likely to perform an action already in its repertoire, as a result of seeing a model perform that action, as well as behaviors such as *mimicry*, *emulation* and *true imitation*. In mimicry, the observer replicates the physical movements of the model, while in emulation it is the end-state generated by the model's actions that is replicated. In true imitation the observer attempts to replicate not only the model's actions, but also their perceived goals.

Learning About Others Information about conspecifics is gathered over the course of interactions. This includes learning the positions of different group members in a *dominance hierarchy*, and *perspective-taking*, where one animal takes actions that take another's visual point-of-view into account. It also includes more advanced *theory of mind*, where one animal must model some aspect on another's internal mental state.

In both comparative psychology and robotics, there has perhaps been too much focus on 'true' imitation, to the exclusion of studying other potentially important and useful social learning mechanisms (this problem is discussed by Byrne and Russon [35] and in accompanying commentaries, including [45], [61],[80] and [99]). In particular, Roitblat [99] notes that there is a danger of defining a phenomena out of existence, by setting the standards at such a level that it cannot be said to occur. He goes on to point out that mechanisms such as *stimulus enhancement*, *goal emulation*, and *response facilitation* may be complex and sophisticated in their own right, and have up until now been used almost exclusively as null hypotheses in the study of imitation in animals.

In a related vein, Call and Carpenter [36] suggest approaching the problem of social learning from a different angle—by looking at the sources of information exploited by the observing animal, rather than focusing on defining the social learning mechanism being used. They suggest that, in any sort of observational learning, three distinct sources of information are available to be observed and copied: the model’s **goals**, the model’s **actions**, and the **results** of those actions, and that each of these information sources provides a different set of useful knowledge about the world. For instance, focusing on the results of actions may help an animal learn about the physical world, and lead to behaviors such as *emulation*, while focusing on the actions themselves may help a creature understand other individuals, and lead to more traditional *mimicry* behavior. Call and Carpenter believe that ultimately, attending to all three sources of information is critical to human social development.

Similarly, it seems likely that, in order to develop socially intelligent robots and animated characters, we will need to implement a variety of mechanisms for taking advantage of the information provided by others, rather than focusing on one form of imitation, or one source of information. As a result, the behavior of the characters in this thesis does not fall precisely into categories such as imitation, stimulus enhancement, or social referencing, but instead allows the characters to combine an amalgam of biologically-inspired abilities in order to correctly reproduce observed behaviors, and begin to identify intentions, motivations and goals. In turn, it is possible that looking at how these sorts of abilities are implemented in an artificial system will give us greater insight into the different cognitive mechanisms behind animal social learning. In the next section, I look at how the perception and production of hierarchical action structures can allow animals (and potentially, artificial creatures) to take advantage of multiple levels of observational information.

2.1.1 Imitation and Hierarchical Action Structures

Hierarchical, motivationally-driven behavior selection mechanisms have frequently been suggested in the animal behavior literature (see for instance [111] and [44] for some classic examples). Timberlake [110] [109] has proposed a particularly detailed

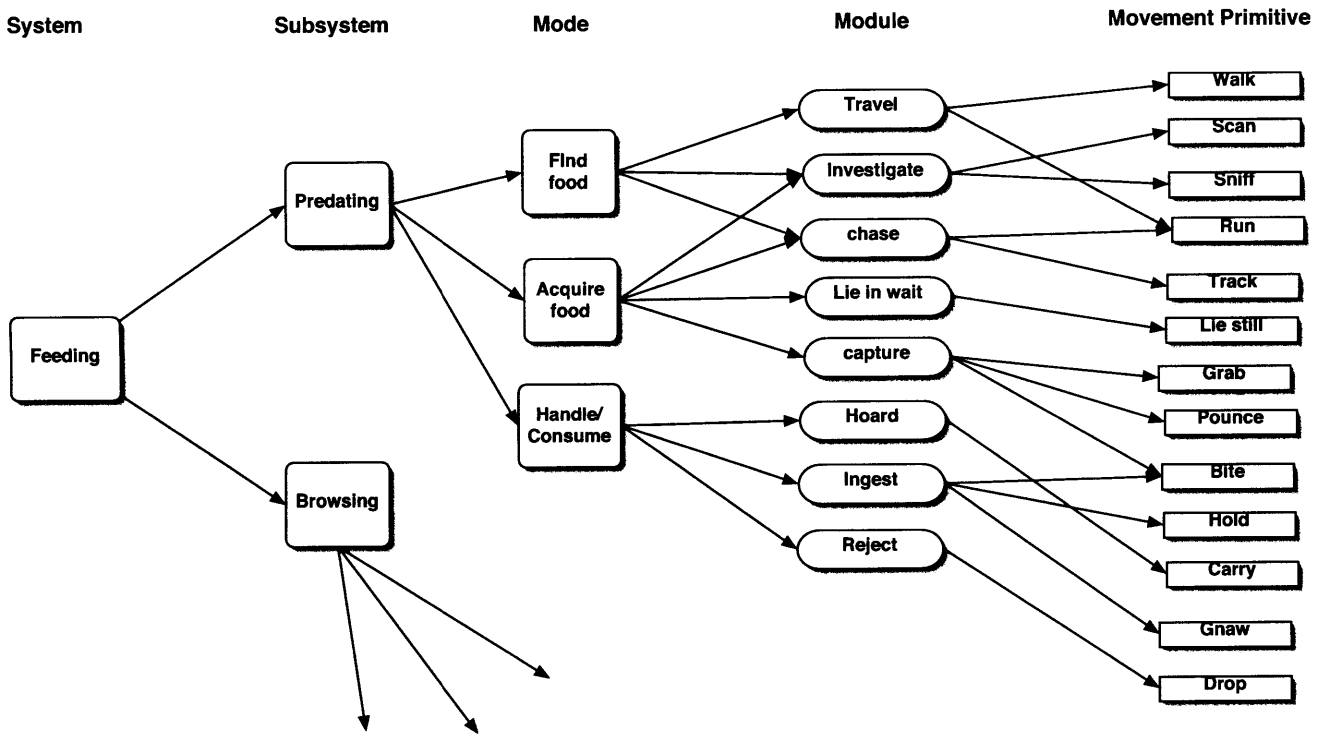


Figure 2-1: An example motivational system for animal feeding (after Timberlake 1989 [110])

theory of hierarchical behavioral structures in animals, known as the *behavior systems* approach. According to Timberlake, an animal’s action hierarchy is composed of behavioral systems, each of which is associated with an innate motivation or drive, such as feeding, self-defense, or socializing. Within a motivational system, each level of the hierarchy contains increasingly specific, sequentially organized actions for satisfying the associated drive (an example motivational system is shown in figure 2-1). This type of action structure is intuitively appealing because it breaks behaviors down into the same sorts of levels and sequences people tend to use when describing a task. Research has shown that people naturally parse action streams into hierarchies of intentional relations [9] [10].

Using the idea of hierarchically organized action systems, such as those proposed by Timberlake, Bryne and Russon [35] have proposed another way in which to broaden the definition of imitation. They suggest that much animal imitation occurs at the “program” level, where an animal with a hierarchical action system learns a program

for organizing its actions by observing the hierarchical structure of another animal’s behavior. Subsequently, it is this hierarchical organization that is imitated, rather than the surface form of the other animal’s movements. *Program level imitation* is contrasted with what they define as *action level imitation*, in which it is the specific physical movements of the model that are replicated. Byrne and Russon suggest that most task-oriented imitation is program-level imitation, whereas action-level imitation is more rare, and may serve a primarily social purpose (this can be seen as somewhat analogous to Call and Carpenter’s discussion of imitation of actions, results, and goals, discussed in the previous section). In addition to Byrne and Russon’s observational studies, some support for this theory comes from Whiten’s experimental demonstrations of imitation of sequential (and potentially hierarchical) action structures in chimpanzees, and imitation of hierarchical behaviors by young children [121].

Byrne and Russon’s theory emphasizes the idea that imitation may operate at a number of levels, and outlines a possible mechanism by which this could occur—the perception and production of hierarchical action structures. They suggest that imitation occurs at multiple stages of the action hierarchy: from imitating individual movement primitives at the lowest level, to imitating the arrangement of behavioral modes and modules (to borrow Timberlake’s terminology), to adopting the high-level goal or motivation at the top of the hierarchy.

Most previous work in robotic imitation has focused on teaching robots or animated characters individual actions meant to solve a particular task, taking advantage of only the lowest level of imitation. Since our behavior architecture is based on a hierarchical action system, we are in an excellent position to explore and take advantage of imitative learning at other levels of the action hierarchy.

2.2 Learning About Others

Note: Portions of the following section are adapted or reprinted from [28].

Research in the field of human cognitive development suggests that the ability to learn

by watching others, and in particular, the ability to imitate, are not only important components of learning new behaviors (or new contexts in which to perform existing behaviors), but could be critical to the development of appropriate social behavior, and ultimately, theory of mind (ToM). In particular, Meltzoff (see for example [86], [83],[84], [85] and [95]) presents a variety of evidence for the presence of imitative abilities in children from very early infancy, and proposes that this capacity could be foundational to more sophisticated social learning, and to ToM. The crux of his hypothesis is that infants' ability to translate the perception of another's action into the production of their own action provides a basis for learning about self-other similarities, and the connection between behaviors and the mental states producing them. I will explore this idea more thoroughly in the following sections.

2.2.1 Understanding Other's Minds

For artificial creatures to possess human-like social intelligence, they must be able to infer the mental states of others (e.g., their thoughts, intents, beliefs, desires, etc.) from observable behavior (e.g., their gestures, facial expressions, speech, actions, etc.). This competence is referred to as a theory of mind [93], folk psychology [57], mindreading [123], or social commonsense [87].

In humans, this ability is accomplished in part by each participant treating the other as a conspecific—viewing the other as being “like me”. Perceiving similarities between self and other is an important part of the ability to take the role or perspective of another, allowing people to relate to, and empathize with, their social partners. This sort of perspective shift may help us to predict and explain other's emotions, behaviors and other mental states, and to formulate appropriate responses based on this understanding. For instance, it enables us to infer the intent or goal enacted by another's behavior—an important skill for understanding other's actions.

2.2.2 Simulation Theory

Simulation Theory (ST) is one of the dominant hypotheses about the nature of the cognitive mechanisms that underlie theory of mind [57] [43]. It can perhaps best be summarized by the cliché “to know a man is to walk a mile in his shoes.” Simulation Theory posits that by simulating another person’s actions and the stimuli they are experiencing using our own behavioral and stimulus processing mechanisms, humans can make predictions about the behaviors and mental states of others, based on the mental states and behaviors that we would possess in their situation. In short, by thinking “as if” we were the other person, we can use our own cognitive, behavioral, and motivational systems to understand what is going on in the heads of others.

From a design perspective, Simulation Theory is appealing because it suggests that instead of requiring a separate set of mechanisms for simulating other persons, we can make predictions about others by using our own cognitive mechanisms to recreate how we would think, feel, and act in their situation—thereby providing us some insight into their emotions, beliefs, desires, intentions etc. We argue that an ST-based mechanism could also be used by robots and animated characters to understand humans and each other in a similar way. Importantly, it is a strategy that naturally lends itself to representing the internal state of others and of the character itself in comparable terms. This would facilitate an artificial creature’s ability to compare its own internal state to that of a person or character it is interacting with, in order to infer their mental states or to learn from observing their behavior. Such theories could provide a foothold for ultimately endowing machines with human-style social skills, learning abilities, and social understanding.

2.2.3 Imitation and Simulation Theory

Meltzoff proposes that the way in which infants learn to simulate others is through imitative interactions. For instance, Meltzoff [84] hypothesizes that the human infant’s ability to translate the perception of another’s action into the production of their own action provides a basis for learning about self-other similarities, and for

learning the connection between behaviors and the mental states producing them.

Simulation Theory rests on the assumption that the other is enough “like me” that he can be simulated using one’s own machinery. Thus, in order to successfully imitate and be imitated, the infant must be able to recognize structural congruence between himself and the adult model (i.e., notice when his body is “like” that of the caregiver, or when the caregiver’s body is “like” his own). The initial “like me” experiences provided by imitative exchanges could lay the foundation for learning about additional behavioral and mental similarities between self and other.

There are a number of ways in which imitation could help bootstrap a Simulation Theory-type ToM [85]. To begin with, imitating another’s expression or movement is a literal simulation of their behavior. By physically copying what the adult is doing, the infant must, in a primitive sense, generate many of the same mental phenomena the adult is experiencing, such as the motor plans for the movement. Meltzoff notes that the extent to which a motor plan can be considered a low-level intention, imitation provides the opportunity to begin learning connections between perceived behaviors and the intentions that produce them. Additionally, facial imitation and other forms of cross-modal imitation require the infant to compare the seen movements of the adult to his own felt movements. This provides an opportunity to begin learning the relationship between the visual perception of an action and the sensation of that action.

Emotional empathy and social referencing are two of the earliest forms of social understanding that facial imitation could facilitate. Experiments have shown that producing a facial expression generally associated with a particular emotion is sufficient for eliciting that emotion [107]. Hence, simply mimicking the facial expressions of others could cause the infant to feel what the other is feeling.

2.2.4 Mirror Neurons

Interestingly, a relatively recently discovered class of neurons in monkeys, labeled mirror neurons, has been proposed as a possible neurological mechanism underlying both imitative abilities and Simulation Theory-type prediction of other’s behaviors

and mental states [124] [53]. Within area F5 of the monkey's premotor cortex, these neurons show similar activity both when a primate observes a goal-directed action of another (such as grasping or manipulating an object), and when it carries out that same goal-directed action [52] [98]. This firing pattern has led researchers to hypothesize that there exists a common coding between perceived and generated actions [94]. These neurons may play an important role in the mechanisms used by humans and other animals to relate their own actions to the actions of others. To date, it is unknown if mirror neurons are innate in humans, learned through experience, or both.

Mirror neurons are seen as part of a possible neural mechanism for Simulation Theory. By activating the same neural areas while perceiving an action as while carrying it out, it may not only be possible but also necessary to recreate additional mental states frequently associated with that action. A mirror neuron-like structure could be an important building block in a mechanism for making predictions about someone else's intentions and beliefs by first locating the perceived action within the observer's own action system, and then identifying one's own beliefs or intentions typically possessed while carrying out that action, and attributing them to the other person.

2.2.5 Understanding Observed Actions

People and other animals often interpret and react to the behavior of others in ways that, at least implicitly, assume that others have intentionality and internal mental state. At the most basic level, when one person watches another, they must divide the continuous stream of motion they observe into individual units of action. Experiments by Baldwin and Baird [10] [9] have shown that, given evidence that they are watching an intentional entity, adults "appear to process continuous action streams in terms of hierarchical relations that link smaller-level intentions (e.g. in a kitchen cleaning-up scenario: intending to grasp a dish, turn on the water, pass the dish under the water) with intentions at higher levels (intending to wash a dish or clean a kitchen)." (p.172, [10]). In other words, adult humans are biased to interpret actions they observe as

part of an intentional or motivational action hierarchy, much like that described in section 2.1.1.

Furthermore, “adults reliably identify certain actions at the more fine-grained level as especially crucial or defining of intentions at the higher level; for instance, the action of scrubbing a dish with a brush is more of a crux for completing the intention to wash a dish than is the equally necessary but less central prior action of turning on the water” (p.172, [10]). This idea, that certain movements can ‘capture the essence’ of an action, is especially important to this work. It suggests that, one way in which a more primitive imitation-based movement identification system can be bootstrapped for more complex social skills, is through the identification of “characteristic” movements, which, when observed, can serve as clues to what the higher level behavior being performed is.

Interpreting observed actions as intentional is not limited to adults. Baird and Baldwin have established that similar abilities exist in infants [9] [10], while Csibra has demonstrated that infants are biased to interpret movements with certain formal structures (e.g. self propelled, following indirect paths, obstacle avoidance) as being goal-directed, even when watching abstract shapes rather than other people or animals [39]. Finally, Meltzoff [83] has shown that by 18 months of age, infants imitate the apparent goal or intention of an action, rather than the action itself (this is demonstrated by their ability to produce the desired result of an action, in response to seeing the action attempted unsuccessfully).

Baldwin and Baird propose that humans use both top-down inferential and bottom-up perceptual mechanisms for dividing observed motion into separate acts. They suggest that intentional behavior is marked by certain predictable features.

For example, to act intentionally on an inanimate object, we must locate that object with our sensors (inanimates do not do this, as they usually do not have sensors). We then typically launch our bodies in the direction specified by our sensors, extend our arms, shape our hands to grasp the relevant object, manipulate and ultimately release it (inanimates usually do not do any of this either). All of this typically coincides with a char-

acteristic kind of ballistic trajectory that provides a temporal contour or ‘envelope’ demarcating one intentional act from the next...

This is all to say that on a purely structural level—the level of statistical regularities—there is considerable information correlated with intentions that is inherent in the flow of goal-directed action. (p.174 [10]).

While there is compelling evidence that these sorts of demarkations and statistical regularities do in fact occur around the boundaries of intentional acts, bottom-up processing alone cannot account for the human ability to interpret observed behavior. This is because “the surface flow of motion people produce in most, if not all, cases is consistent with a multitude of different intentions” (p.175 [10]). In other words, one can walk towards an object, or even pick it up, for any number of reasons. In order to decide between competing candidate intentions, humans must turn to other sources of knowledge, potentially including their own behavior systems.

Intentionality in Animals

Although there is more controversy surrounding the extent to which non-human animals understand intentionality, there is evidence that they can at least take advantage of information about another animal’s point-of-view. At the simplest level, many primate species demonstrate gaze-following behavior [112]. Chimpanzees in particular are able to follow human gaze around obstacles and past distractors, adjusting their position and checking back with the gazing model repeatedly to determine their object of attention [115]. Similarly, when tested in a competitive setting, chimpanzees have been shown to understand what other chimpanzees can and cannot see, and make judgments about which food sources to pursue accordingly [58] [59] [113].

While it is not entirely clear whether these types of behaviors result from a primitive theory of mind, Whiten [120] makes a compelling argument for where the transition between behavior-reading and mind-reading begins. Whiten posits that, at their simplest, mental-states can be seen as intervening variables between observable behaviors. Figure 2-2 gives an example of the use of an internal variable. In a case such

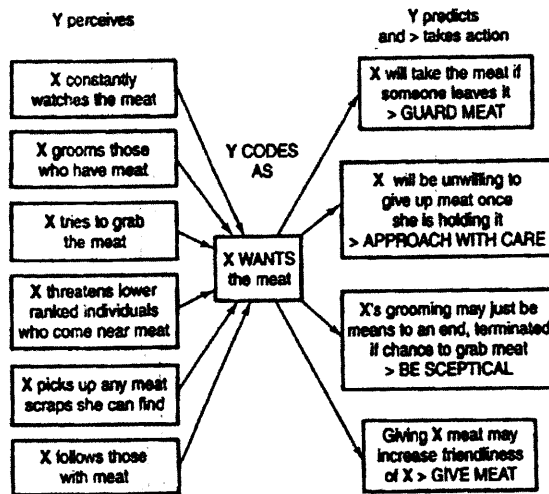


Figure 2-2: This is a hypothetical example, where primate Y watches primate X. By positing an internal state of ‘wanting’ in primate X, Y can gain economy of representation (from Whiten 1996 [120]).

as in figure 2-2, there are enough potential cause and effect behaviors, that positing an internal variable can be very beneficial to the animal, allowing it to avoid having to learn each of the different individual cause and effect links. An animal using such an internal variable in its interpretation of another’s action could be said to have moved from behavior-reading to mind-reading. Whiten therefore suggests that mind-reading becomes a useful strategy exactly at the point where behavior is so complex and varied that it is difficult to interpret *without* positing any intervening invisible states or variables.

2.2.6 From Social Animals to Social Characters

The cognitive literature provides compelling evidence for the presence and usefulness of social learning in human and non-human animals. Furthermore, a number of themes can be seen in the animal behavior and infant development studies described here, which can be used to guide our design of socially capable artificial creatures.

Multiple Levels of Social Learning. Social learning and imitation may happen at many levels of behavioral granularity.

Multiple Sources of Information. There are multiple sources of information contained in an action, and each provides opportunities for different kinds of social learning.

Motivationally-driven Action Hierarchies. One possible way in which to represent multiple sources of information and multiple levels of behavioral granularity is by using a motivationally-driven hierarchical action structure.

Simulation Theory. Simulation Theory, where the creature uses itself to help interpret another's behavior, may be an especially useful approach to developing social abilities.

Perception-Production Coupling. A Simulation Theory-style social learning system may have a perception-production coupling mechanism, such as mirror neurons, behind it.

Bootstrapping from Imitation. Being able to identify and imitate another's behavior may be the first step towards more complex interpretation of that behavior.

With these points in mind, the goal of this thesis can now be further refined. The aim of this work is not only to create synthetic characters capable of social learning, but to do so using a cognitively inspired approach. In particular, we would like to explore the mechanisms by which Simulation Theory can be used by one character to learn from another's behavior. Our implementation of a simulation-theoretic social learning system will take advantage of the hierarchical action system used by our creatures, and attempt to exploit the multiple levels of social learning and multiple sources of observational information available. Finally, we will use the ability to recognize and reproduce observed movements as the starting point for developing more complex social skills, such as identifying simple motivations and goals, and learning about objects in the environment.

Chapter 3

Background and Related Work

Note: Portions of this chapter are reprinted or adapted from [28].

The fascinatingly rich array of animal social behavior has provided frequent inspiration to the artificial intelligence community. Some of the first research into multi-agent systems has occurred in the fields of swarm intelligence [24] and distributed robotic systems [96], which draw inspiration from the complex societies of social insects. Similarly, bird flocking behavior gave rise to Reynolds' now classic Boids [97], along with related works such as Tu's modeling of schools of fish [117].

In their comprehensive review of socially interactive robots, Fong *et al.* [51] point out that what is common to approaches such as those described thus far, is that the individual participants are anonymous and interchangeable. The group behavior is self-organizing, and does not require individuals to differentiate between each other, learn from each other, or form individual relationships. Fong *et al.* call this type of agent or robot "group social" and distinguish them from "individual social" robots which are defined as:

embodied agents that are part of a heterogeneous group: a society of robots or humans. They are able to recognize each other and engage in social interactions, they possess histories (perceive and interpret the world in terms of their own experience), and they explicitly communicate with and learn from each other. (p.144 [51]).

This thesis is concerned with working towards exactly this sort of socially intelligent artificial creature—characters able to learn about each other and their environment through social interactions. In this section, I will highlight related work in developing socially interactive characters and robots, focusing especially on prior work in imitative artificial creatures.

3.1 Social characters and robots

3.1.1 Interactive Animated Characters

Much of the previous work done by the Synthetic Characters Group has focused on creating animated synthetic creatures who interact with humans and each other in a compelling and believable manner. Some of the first work in ethologically inspired interactive characters was pioneered by Blumberg [18]. The cognitive architecture developed by Blumberg was used in the ALIVE installation [22], where human participants saw themselves projected into a virtual world featuring an animated dog named Silas. The ALIVE system was able to recover the locations of the participant’s head, hands and feet, and could also do simple gesture recognition, allowing Silas to respond to gestural commands for behaviors such as *sit* and *shake*. Silas could also interact with the participant based on his own motivations, for instance by bringing a ball to the person’s hand if his desire to fetch was high. A major contribution of this work was to introduce the idea of a hierarchical action system composed of competing motivational subsystems as a behavior selection mechanism for autonomous characters, an idea inspired by the ethological literature (as discussed in section 2.1.1).

In more recent work, the group created the Alphawolf installation [116] (shown in figure 3-1), a project focusing on the social dynamics of a semi-autonomous wolf pack. In Alphawolf, human participants could assume the role of one of three wolf pups, and could control the pup’s actions by howling, growling, whimpering or barking into a microphone. Based on their human-influenced interactions with each other,

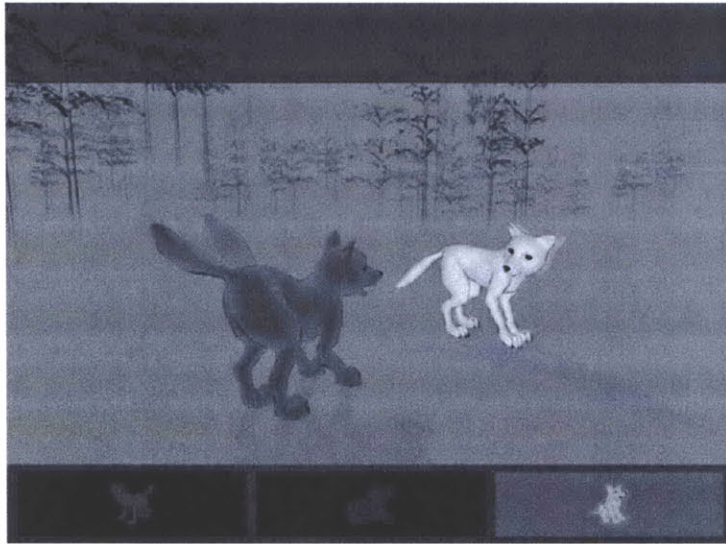


Figure 3-1: two wolf pups in the Alphawolf installation interacting.

the pups would form a dominance hierarchy, and develop emotional memories of one another, representing how dominant or submissive they felt towards the other pups. This project was one of the first to explore enduring social memories in artificial characters, and continued the Synthetic Characters tradition of exploring human control of intelligent artificial creatures [68] [20].

Other Systems

Other researchers have also addressed the problem of creating believable and life-like animated characters. These include Perlin and Goldberg's pioneering IMPROV system [91], which allows synthetic actors to move naturally in response to relatively high-level human direction. Badler and colleagues have done significant research developing 3D characters capable of executing complex actions in response to natural language instructions. To this end, they have developed the Parameterized Action Representation (PAR), meant to be the conceptual bridge between natural-language instructions and carrying out a particular action [8]. Besides the action itself, a PAR consists of the agent meant to carry out the action, conditions under which the action may be performed, expected results of the action, possible subactions, and objects to perform the action on—it can thus be seen as somewhat analogous to the

action tuples used in our system and described in section 5.6. Badler’s group has also developed numerous other tools for expressive character animation, including the EMOTE system, which allows the emotional appearance of a character’s motion and facial expressions to be easily modified [7].

While both Badler and Perlin’s systems contain a strong set of tools for creating expressive characters, the focus of both these works is primarily on life-like appearance rather than life-like cognition, whereas this thesis is more concerned with the latter, in the hope that it leads to the former.

3.1.2 Social Robots

Over the years, a number of robotic systems have been designed for the express purpose of exploring human-robot social interactions, and perhaps the most well known of these is Breazeal’s Kismet [25] [27](shown in figure 3-2). Built to model the

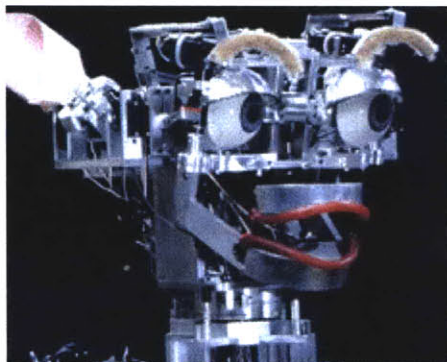


Figure 3-2: Kismet.

interactions between human infants and their caregivers, Kismet has an expressive high degree of freedom face, with exaggerated, cartoon-like features, and no body. Kismet’s baby-like appearance, as well as its behavior, was specifically designed to both encourage and take advantage of the kinds of social exchanges that human infants and their caretakers typically participate in. In particular, Kismet engages users in turn-taking games and conversations, has human-like emotional responses to tone of voice, and can regulate interactions with humans by altering its level of engagement—slowing down or withdrawing from the interaction if it became over-

stimulated, or seeking out human attention when left alone. At the heart of Breazeal's Kismet research is the idea (discussed in section 2.2.3) that the caregiving behaviors people offer infants act as a scaffolding for infant social learning. Robots that elicit these same sorts of behaviors might similarly be able to use them to bootstrap better understanding of the humans in their world.

In a similar vein, Scasselati's work with the robot Cog [102] looked at how different theories of theory of mind presented in the cognitive literature could be implemented in a humanoid robot. Much of Scasselati's research focused on implementing the low-level abilities suggested by the cognitive literature, such as face tracking and recognition, as well as more advanced skills such as gaze-following and joint attention.

Roy's research has focused on creating interactive robots whose understanding of the world around them helps ground their understanding of natural language. The most sophisticated of these robots to date is Ripley [100] (pictured in figure 3-3), a 7 degree of freedom robot capable of a variety of complex linguistic interactions with human users. In the context of developing sophisticated language abilities, a number of mental models for representing the people and objects in its environment have been developed for Ripley [101].

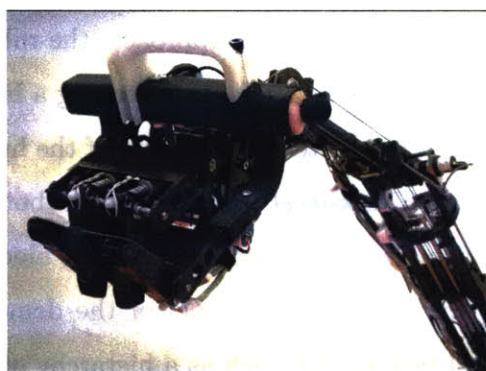


Figure 3-3: The robot Ripley (image taken from www.media.mit.edu/cogmac)

3.2 Imitative characters and robots

In recent years, a number of robotic and animated characters, with a variety of imitative abilities, have been developed (for a partial review see [31] and [104]), some of them using biologically inspired approaches [28] [30] [77] [46] [12]. Often, this work has emphasized the creation of systems able to mimic the particular form of individual actions, focusing on the physical performance of the robot or character, rather than on gaining social or environmental knowledge. Some researchers have focused on using imitation to allow robots to learn interpersonal communication protocols, either from other robots or from human instructors [13] [11]. However, this work has taken relatively little advantage of other types of social learning that are present in animals, especially with regard to possible shared mechanisms between simpler social learning behaviors and mind-reading abilities. Overall, the path towards creating socially intelligent agents is still largely uncharted, especially within the framework of a larger cognitive architecture.

3.2.1 Learning new actions

Many imitative systems have been designed with the aim of using imitation to constrain the problem of learning which actions to perform in what situations, a problem generally termed state-action space discovery. Some of the earliest work in this area is called learning by demonstration. In this approach, the robot (often a robotic manipulator) learns how to perform a new task by watching a human perform the same task. This may or may not involve literal mimicking of the human’s behavior. In the case where it does not, called *task-level imitation*, the robot learns how to perform the physical task of the demonstrator—such as stacking blocks [71] or peg insertion [63]—without imitating the specific movements of the demonstrator. Instead, the robot acquires a high-level task model, such as a hierarchy of goal states and the actions to achieve them, from observing the effects of human movements on objects in the environment. *Task-level imitation* can be seen as somewhat analogous to the processes of *emulation* or perhaps *program-level imitation* seen in animals (discussed in

chapter 2). The characters implemented in the context of this thesis are also capable of picking out the goal-states of others, however this implementation uses Simulation Theory, rather than simply observing and replicating changes to the environment (for further discussion, see chapter 6).

In other work with highly articulated humanoid robots, learning by demonstration has been explored as a way to achieve efficient learning of dexterous motor skills [5] [103]. The state-action space for such robots is too prohibitively large to search for a particular solution in reasonable time. To address this issue, the robot observes the human’s performance, using both object and human movement information to estimate a control policy for the desired task. The human’s demonstration helps to guide the robot’s search through the space, providing it with a good region to initiate its own search. If given knowledge of the task goal (in the form of an evaluation function), robots have learned to perform a variety of physical tasks—e.g., learning the game of “ball in cup” , or a tennis forehand [89] [88], by utilizing both the demonstrator’s movement and that of the object. We are also interested in the problems of action and state-action space discovery, and think imitation is a very worthwhile approach to this task. However, the main focus of this work is on developing characters with a better understanding of others and of their environment, rather than on the learning of novel actions (for previous work by the Synthetic Characters Group on action and state-action space discovery see [21]).

Another way to accelerate learning is to encode the state-action space using a more compact representation. This makes the overall state-action space smaller and therefore faster to explore. Researchers have used biologically-inspired representations of movement, such as movement primitives [17] [78], to encode movements in terms of goal-directed behaviors rather than discrete joint angles. Primitives allow movement trajectories to be encoded using fewer parameters and are combined to produce the entire movement repertoire. The tradeoff of this compact representation is loss of granularity and/or generality of the movement space. As a result, more recent work has focused on using imitation as a way of acquiring new primitives (as new sequences or combinations of existing primitives) that can be added to the repertoire [67] [50].

As will be discussed in section 5.7 and chapter 6, the system described in this thesis also incorporates the idea of movement primitives and goal-directed behaviors, which can be added to and adapted at run-time.

Recently, Lieberman [75] developed a system for teaching a humanoid robot dexterous motor skills through human demonstration, and for automatically parsing observed motion streams into individual skills. Lieberman’s system takes an intentional approach to action parsing, analyzing motion and end-effector position with respect to objects in the environment, in order to find the intentional boundaries of movements. His system is also able to develop new motion spaces by correctly editing and combining multiple differing examples of a task-oriented movement (e.g. picking up a cup from a number of different angles) and interpolating between them (for more information on motion spaces created by interpolating multiple animations, see the discussion of blended animations in section 5.7.3).

3.2.2 Learning to imitate

In learning to imitate, the robot learns how to solve what is known as *the correspondence problem* through experience (i.e., how to map the observed movement of another onto the character’s own movement repertoire). One strategy for solving the correspondence problem is to represent the demonstrator’s movement trajectory in the coordinate frame of the imitator’s own motor coordinates. This approach was explored by Billard and Schaal [14] who recorded human arm movement data using a Sarcos SenSuit and then projected that data into an intrinsic frame of reference for a 41 degree-of-freedom humanoid simulation.

Another approach, the use of perceptual-motor primitives [118] [67], is inspired by the discovery of *mirror neurons* in primates. These neurons are active both when a goal-oriented action is observed and when the same action is performed (recall section 2.2.4). Mataric [77] implements this idea as an on-line encoding process, that maps observed joint angles onto movement primitives to allow a simulated upper torso humanoid to learn to imitate a sequence of arm trajectories.

Others have adapted the notion of mirror neurons to predictive forward models

[126]. For instance, Demiris and Hayes [46] present a technique that emphasizes the bi-directional interaction between perception and action, where movement recognition is carried out by the movement generating mechanisms. To accomplish this, a forward model for a behavior is built directly into the behavior module responsible for producing that movement. In model-based imitation learning, the imitator's motor acts are represented in task space where they can be directly compared with the observed trajectory. Using this approach, Atkeson and Schaal [4] show how a forward model and a priori knowledge of the task goal can be used to acquire a task-level policy from reinforcement learning in very few trials. They demonstrated an anthropomorphic robot learning how to perform a pole-balancing task in a single trial and a pendulum swing up task in three to four trials [4] [5].

As discussed in section 2.2.4, our implementation is also inspired by the possible role that mirror neurons play in imitative behavior. In the approaches described above, mirror neuron-inspired mechanisms are an on-line process for either mapping perceived movements to another coordinate frame, or are a forward model that is directly involved in generating the observed action. In contrast, our implementation is consistent with that discussed in Oztop and Arbib [90] and Meltzoff and Decety [85], where mirror neurons are believed to represent observed movement in terms of the creature's own motor coordinates. This concept of explicit representation (i.e. memory) is important in order to capture the goal-directed match-to-target search that characterizes exploratory imitative behavior of infants [87]. It is also important in order to account for the ability of young infants to imitate deferred actions after a substantial time delay (on the order hours and even days) that Meltzoff has observed [81] [82] .

Finally, the correspondence problem has also been addressed in the animation and motion capture literature, where it is known as the problem of *retargetting* [55], or taking motion capture or animation data from one character and using it to animate another character of differing size. Some particularly interesting work in this area has been done by Bindiganavale [16], whose CaPAR system parses motion capture data into hierarchical goal-directed action units, which can then be played out on a

new character after only a single example. However, this work is approached from a very different vantage point than our own, since its goal is primarily to allow more flexible use of motion capture or animation material, rather than on giving synthetic characters additional cognitive functionality. As a result, this approach does not take advantage of already existing cognitive mechanisms within the characters, who are viewed primarily as directed actors rather than independent agents.

3.2.3 Learning by imitation

Imitative behavior can either be learned or specified a priori. In learning by imitation, the robot is given the ability to engage in imitative behavior. This serves as a mechanism that bootstraps further learning and understanding from guided exploration by following a model. Initial studies of this style of social learning in robotics focused on allowing one robot to learn reactive control policies to navigate through mazes [60] or an unknown landscape [41] by using simple perception (proximity and infrared sensors) to follow another robot that was adept at maneuvering in the environment. This approach has also been applied to allow a robot to learn inter-personal communication protocols between similar robots, between robots with similar morphology but which differ in scale [12], and with a human instructor [11].

Learning by imitation often advocates an “empathic” or direct experiential approach to social understanding whereby a robot uses its internal mechanisms to assimilate or adopt the internal state of the other as its own [41] [70]. Given our discussion in section 2.2.1, we also advocate a simulation theoretic approach to achieve social understanding of people by robots and animated characters. However, a pure empathic understanding where the character simply “absorbs” the experience, and does not distinguish it as arising from self, or being communicated by others, is not sufficient for human-style social intelligence. For many forms of social learning, the character must be able to determine what is held in common and what is not—these include social referencing, and cooperative and competitive activities, where separating your own knowledge from the knowledge of the other is especially paramount. In our approach, the character can use its own cognitive and affective mechanisms as a “simulator”

for inferring the other’s internal states, which are represented as distinct from the character’s own states.

3.2.4 Towards Socially Intelligent Characters and Robots

Recently, a number of projects by the Robotic Life Group at the MIT Media Lab have begun addressing the problem of creating socially intelligent robots, focusing especially on the development of robots who can cooperate with humans. In particular, the group’s expressive humanoid robot, Leonardo, can learn a number of collaborative button-pressing tasks, using human guidance to quicken the learning process [29]. The robot forms and refines hypotheses about the goals of the task by listening and responding to verbal instruction, observing human gesture (e.g. pointing), and looking at changes in the environment (e.g. which buttons have changed state) [76]. Subsequently, the robot can perform the task collaboratively with the human, completing some of the task goals, while allowing the human to complete others. Throughout the interaction, the robot uses communicative gestures to aid the learning and collaboration processes. Similarly, Leonardo can also be taught games to play with the human instructor [32]. In the case of competitive games, this requires Leonardo to know that the human’s goal in the task is different than his own, necessitating an explicit representation of self and other’s goals.

In other work with the robot Leonardo, we created a cognitively-inspired facial imitation architecture. Our implementation was heavily inspired by the imitative interactions human infants and their caregivers frequently engage in [84], and by the *Active Intermodal Mapping* (AIM) theory of facial imitation proposed by Meltzoff and Moore [87]. AIM suggests that a combination of innate knowledge and specialized learning mechanisms underlie infants ability to imitate. Specifically, AIM proposes that infants have an innate ability to recognize other’s facial organs, and that they map their own movements, and the movements they observe, onto the same internal representation (hence, *intermodal mapping*). In other words, AIM presents a model for the implementation of Simulation Theory via perception-production coupling in infant facial imitation (for more details on the AIM model, see for example Meltzoff



Figure 3-4: An animated version of the Leonardo robot imitates a human model's facial expressions.

and Moore 1997 [87]).

In this implementation, an animated version of Leonardo is able to learn the correspondence between his own facial features and those of a human model by having the human imitate him. Subsequently, he is able to imitate the human's facial expression (shown in figure 3-4). This ability was then used to help bootstrap social referencing capabilities in the robot, where the robot used the emotions typically associated with its own facial expressions to judge the emotional stance of a human model towards objects in the environment, and respond accordingly.

The implementation of facial expression mimicry in Leonardo was an important first step towards creating a robot with Simulation Theory-style social learning abilities. It demonstrated that social learning skills, such as social referencing, could be bootstrapped from facial imitation abilities. However, in this implementation, Leonardo's imitative abilities were limited to static facial expressions, and the Simulation Theory he employed occurred only at the level of his movement primitives. This thesis takes many of the ideas introduced in that work further, creating a social learning system that applies Simulation Theory at the level of goal-directed actions as well as movements, allowing the characters presented here to imitate whole gestures and movements, identify simple motivations and goals for other's actions and learn about objects in the environment by watching others interact with them. In the next chapter, I will use the test characters Max and Morris Mouse (who will be the focus of the remainder of this thesis) to present the full spectrum of social learning abilities implemented in this thesis.

Chapter 4

Max and Morris

Max the Mouse and his friend Morris (pictured together in their virtual environment in figure 4-1), are two anthropomorphic animated mouse characters, and the testbed for the Simulation Theory-based social learning system presented in this thesis. Max



Figure 4-1: Max (right) and Morris (left) in the virtual desert

and Morris inhabit a rather minimalist graphical world, populated only by themselves, and sometimes containing a small number of simple objects representing food and toys (see figure 4-2) They star in a number of small interactive demonstrations, meant

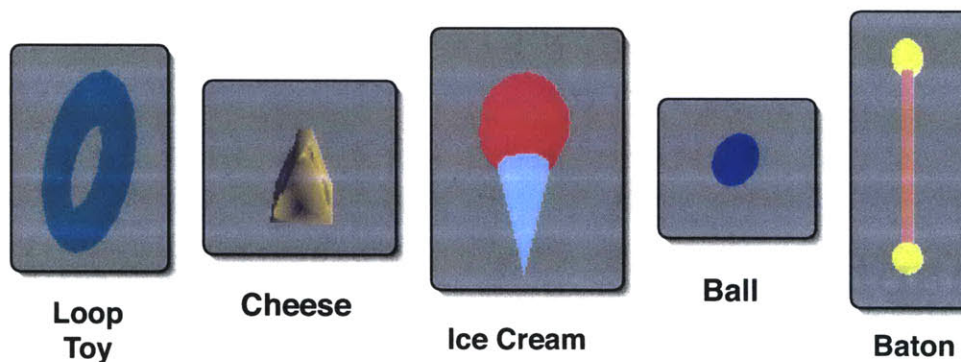


Figure 4-2: The objects that can be introduced into the world.

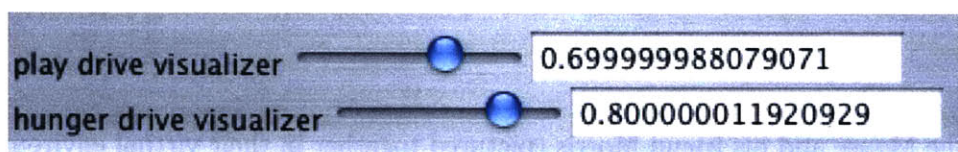


Figure 4-3: Visualizer showing the current level of Max's hunger and play drives

to test and exhibit the imitative and social learning capabilities implemented in this work. In this chapter, I will present an overview of these demonstrations, and describe Max and Morris's behavior, in order to ground the discussion of architecture and implementation in the remainder of the thesis. Videos of interactions similar to the sample interactions described below will be available at

www.media.mit.edu/~daphna/thesis_videos.html

4.1 Sample Interaction :The Basics

Max and Morris start out facing each other in the middle of a barren desert (they are the last of the little known sahara mouse species *Unluckius rattus*). A *Drive* visualizer visible on screen shows that Max's levels of playfulness and hunger are both high (pictured in figure 4-3), and the same is true for Morris. However, there are no food or toy objects available in the environment (as of yet, there are no objects in the environment at all), and Max and Morris begin to pace.

As shown in figure 4-2 there are a small number of objects which can be intro-



Figure 4-4: Buttons for introducing objects into the world



Figure 4-5: Max (left) and Morris (right) jumping for a piece of cheese

duced into the world, by choosing among corresponding labeled buttons (figure 4-4). Pressing any of the buttons causes an object of that type to appear in the world, hovering over the mice's heads, and a corresponding array of sliders, which manipulate the object's location, to appear in the button panel. In this case, a piece of cheese is introduced.

As soon as the piece of cheese is added to the world, both mice stop pacing and navigate over to it. They then begin jumping for the cheese (seen in figure 4-5). As the cheese is manipulated — lowered down, moved sideways — they adjust their jumping direction and orientation, and begin reaching instead of jumping when the

cheese is closer to them. Eventually, the cheese is moved so that it is close enough for Max to get it. Max eats the cheese, it disappears and Max's level of hunger in the *Drive* visualizer drops. While Max was eating the cheese, Morris continued reaching for it, but he stops reaching for it once it is gone.

Next, a ball is added to the world, and both mice begin to reach for it. However, after a new piece of cheese is also introduced, Morris, whose level of hunger is still high, switches to jumping for the cheese, while Max continues jumping for the ball. Finally, after some time has elapsed, Max's level of hunger is high again, and he too begins to reach for the cheese.

4.2 Sample Interaction: Imitation

As before, Max and Morris start out facing each other in an otherwise empty world. This time, no objects are introduced, and instead Max is instructed to observe Morris. Max orients towards Morris and, as he does so, a graphical window displaying the world from Max's perspective shows a stick figure version of Morris coming into view (figure 4-6). In the window that represents Max's viewpoint a number of Morris's joints, such as his hands, nose and feet, are marked with colored spheres.



Figure 4-6: Morris, rendered from Max's point of view. Certain key effectors, such as Morris's hands, nose and feet, are marked by colored spheres

There is also a panel of buttons available, labeled with a variety of possible movements such as *wave*, *poundGround*, *coverEyes*, *jump* and *thumbsUp* (figure 4-7). The

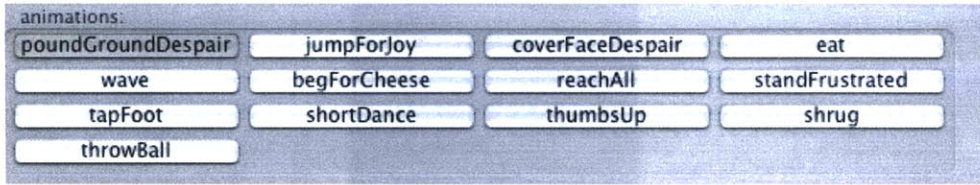


Figure 4-7: Buttons for requesting that Morris perform different movements

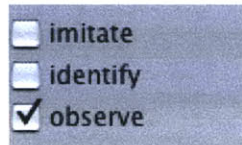


Figure 4-8: Buttons for requesting that Max observe, imitate or identify Morris's action

version of Morris used in this interaction is somewhat different than the previous Morris—he must perform any movement chosen from the button panel. As described in appendix B, a movement is simply an individual motion primitive or gesture. In this case, the button labeled *jump* is chosen, and as Morris jumps up into the air Max adjusts his gaze so that Morris stays in view.

Once Morris has landed Max is asked to imitate the movement he last saw Morris perform (see figure 4-8). Based on his observation of Morris, Max finds the movement he knows how to perform that is closest to what he saw, and begins to jump into the air. Next, Morris is told to cover his face. As Morris raises his arms to cover his face, a number of the colored spheres marking his joints become obscured from Max's perspective (see figure 4-9). Nevertheless, Max correctly reproduces the movement when asked to imitate it (figure 4-10).

4.3 Sample Interaction: Action Identification

Once again, Max and Morris face off in the desert. As in the first scenario, they both have high levels of playfulness and hunger. A ball is added to the world and Max is instructed to observe Morris, who begins jumping for the ball. This time, Max is asked to identify the action Morris is performing, which he does by also beginning to jump for the ball.

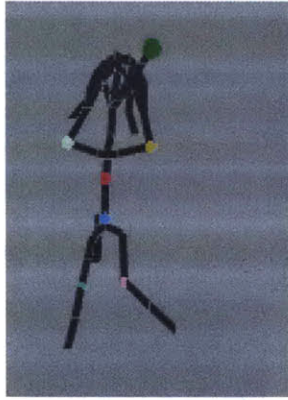


Figure 4-9: Morris covering his eyes, as seen by Max. Notice that some of the spheres marking his body parts are not visible (compare to figure 4-6)

While a movement is an individual motion primitive, an action is a movement or series of movements placed in an environmental and motivational context. Movements represent stand-alone physical motion, while actions are behaviors performed in response to environmental triggers, motivational drives and desired goals states (see appendix B for more details). Max identifies Morris's jumping action, by looking for actions in his own behavior system that use the jumping movement. In this case, he finds that jumping is used to get objects, and that toys such as balls satisfy his play drive. By jumping for the ball, Max shows that he knows that Morris is trying to get the ball, and that he is doing this in order to satisfy his play drive.

A second object is added to the environment, a piece of cheese. The cheese is lowered down closer to Morris and Morris stops jumping for the ball and begins reaching for the cheese. Max is again asked to identify Morris's action, and begins reaching for the cheese (not the ball) as well. By doing so, Max indicates not only that he knows Morris is reaching, but that he is trying to get the cheese because his level of hunger is high.

Now, the cheese is brought even closer to Morris so that he is able to reach it and eat it. When asked to identify Morris's action, Max mimics eating, correctly identifying Morris's action even though the cheese, having been eaten by Morris, is no longer there.

4.4 Sample Interaction: Learning About Objects I

In this interaction an object we haven't seen before is introduced, an ice cream cone. Morris begins to reach for the ice cream cone, but Max, who doesn't 'know' that ice cream is edible (or useful for anything at all), does not attend to the ice cream.

Once again, Max is asked to observe Morris, and identify his action. Max shrugs, indicating that he doesn't know why Morris is reaching (since, to him, the ice cream has no purpose). The ice cream is given to Morris, who eats it. Max is again asked to identify Morris's action, and this time, he mimes eating. Now, when another ice cream cone is added to the environment, Max immediately orients to it and begins reaching for it. When he is given the ice cream he eats it, having learned that ice cream is edible.

4.5 Sample Interaction: Learning About Objects II

While there is only one way for Max and Morris to eat food—by consuming it, there are a number of ways in which they play with toys. When Max or Morris is given a ball, they toss it up and down in the air. When one of them is given a baton, they dance in a circle with it. Both these actions reduce their level of playfulness.

Here, another new object is introduced, a cube-shaped tossing toy. As with the ice cream, Morris knows how to use this toy, but Max does not know its purpose. Morris is given the cube, and begins tossing it, while Max is again told to observe and identify his actions. Subsequently, when Max's level of playfulness is high, he reaches for the cube, and when he is given the cube he begins tossing it. Here, Max has learned not only that he can play with the cube, but *how* to play with it.

4.6 Looking Ahead

In this chapter, I introduced Max and Morris Mouse, and described some typical interactions and behavior patterns for them. In the next chapter, I will describe the overall cognitive architecture underlying their behavior, while in chapter 6 I'll go into the details of their implementation.



Figure 4-10: First row. Morris (blue) demonstrates an action (covering his eyes) while Max (brown) watches. Second row: Morris through Max's eyes. The colored spheres represent key effectors. Third row. Max reproduces Morris's action, by performing the movements in his own repertoire that are closest to what he observed.

Chapter 5

Cognitive Architecture

Max and Morris Mouse are the latest in long line of interactive animated characters developed by the Synthetic Characters Group at the MIT Media Lab [68] [20] [116] [21] [66]. They were built using the Synthetic Characters C5m toolkit, a specialized set of libraries for building autonomous, adaptive characters and robots. The toolkit contains a complete cognitive architecture (or “virtual brain”) for synthetic characters (diagrammed in figure 5-1), including perception, action, belief, motor and navigation systems, as well as a high performance graphics layer for doing Java-based OpenGL 3D Graphics. All of the work described in this thesis was implemented under the C5m system.

Several thorough introductions to the Synthetic Characters toolkit have already been written [21] [34] [65], so I will only present a relatively brief introduction here, focusing particularly on functionality added especially for this work, and on the implementation details of the characters and architectures specific to this thesis.

5.1 System Overview

Figure 5-2 presents a high-level overview of the C5m system. The central component is the *World*, which represents the “ground-truth” of the environment the characters exist and operate in. The *World* keeps track of all the creatures and other physical objects in the synthetic environment, and coordinates all communication and data

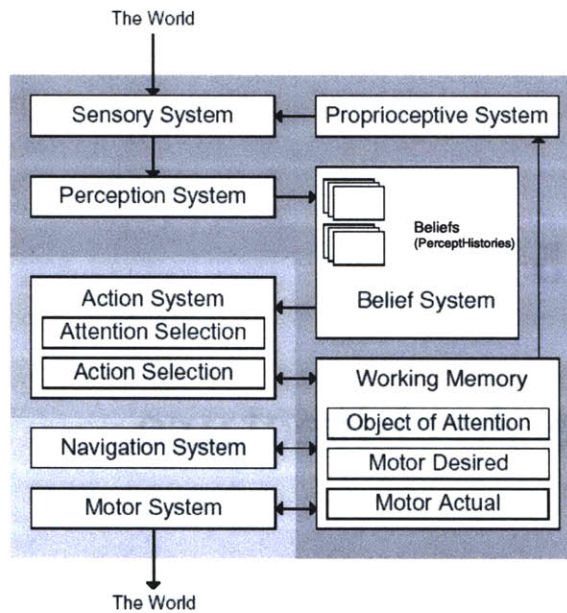


Figure 5-1: This is the overall cognitive architecture used for characters such as Max and Morris. The systems are processed serially, in roughly top-to-bottom (or light-to-dark) order, relative to this figure. Imitation, and other social skills, make particular use of the perception, action and motor systems. (Adapted from Burke 2001 [34])

transfer between creatures, objects, and input devices. Input devices are anything that provides sensory information, including standard interfaces such as keyboards, mice and joysticks, as well as optional microphones, cameras and other physical sensors (e.g. pressure sensors for a robot). Creatures and other objects are also input devices, since they provide sensory information to each other. The *World* and all its objects are optionally hooked up to a graphical front-end, however objects in the synthetic world are not required to have a graphical representation.

5.2 Input to the System

The *World* collects messages called *data records* from input devices, and distributes them to creatures. *Data records* are used extensively throughout the system, and represent sensory information that the creatures may perceive and act upon. For example, a *data record* might contain symbolic visual information, such as a creature's

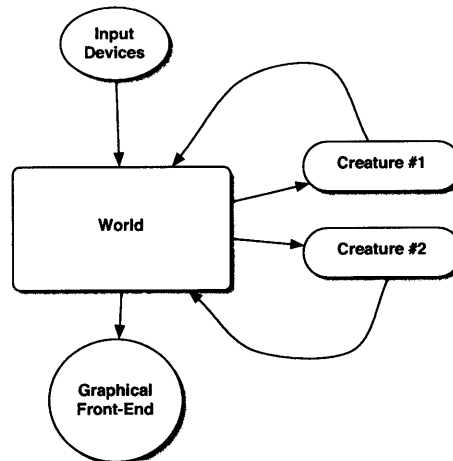


Figure 5-2: An overview of the Synthetic Character's System (After Isla 2001 [65])

current position and “shape”, which each creature posts every timestep. Alternatively, it could contain user-input information that the creatures then “perceive” (e.g. button presses, auditory input). *Data records* are also produced internally by *sensors* in the creature's *sensory system* (described in section 5.3), allowing passively collected sensory information handed in by the *World* (e.g. symbolic vision) and actively collected sensory information sensed by the creature (e.g. synthetic vision, described in section 5.3.1) to be processed by the same perceptual mechanisms down the line.

5.3 Sensory System

A creature's *sensory system* is composed of *sensors*, which are responsible for gathering and filtering a particular type of sensory data. The simplest of these are *input sensors*, each of which is paired with an input device. Every timestep, the *World* collects *data records* from each input device, and makes them available to any *input sensor* registered to receive *data records* from that device. The *input sensor* then filters these records to enforce sensory honesty. Specifically, the *input sensor* filters out sensory information that should not be perceivable by the creature, for example visual events that occur behind the creature. *Sensors* may also perform additional processing on the data, such as low-pass filtering of position information, or even performing sophisticated pattern recognition algorithms on video input.

Another kind of sensory information worth mentioning is proprioceptive input. *Proprioceptive sensors* assimilate information about the creature’s current state, which has been posted to *working memory* (a sort of internal blackboard), by the creature’s other systems. Of particular importance to the work in this thesis is proprioceptive body pose information—the creature’s sense of where in space its body parts are currently located.

5.3.1 Synthetic Vision

For this thesis, another kind of *sensor*, the *synthetic vision sensor*, was implemented (the implementation is almost identical to that used by Isla, and described in [65]). Learning by observation is an inherently visual process, and using synthetic vision, where the character “sees” the world graphically rendered from its own perspective, forces us to grapple with the problems of gesture and movement recognition in a more honest and biologically plausible manner than in a system that uses only symbolic visual information.

The Synthetic Characters group has used synthetic vision in a number of previous projects [19] [66]. For my thesis, I used a simple form of color-coded synthetic vision, shown in figure 5-3. This type of Synthetic Vision has been used previously, for example in [117]. Each timestep, the *synthetic vision sensor* takes as input a graphical rendering of the world from the position and orientation of the creature’s eye. This rendering is typically a color-coded view of the world, in which each object is assigned a unique color, which can be used as an identifying tag. By scanning the visual image for pixels of a particular color, the creature can “see” an object. However, just as with real vision, objects that are obstructed or out of view cannot be seen.

Besides determining whether an object is obstructed or visible, the other important function of the *synthetic vision sensor* is to determine object location. A simple approximate location can be extracted visually by examining the screen-space coordinates of an object’s centroid in the point-of-view rendering. The (x, y) screen-space coordinates can then be combined with information from the rendering’s depth-buffer, in order to determine the location of the object in the coordinate frame of the crea-

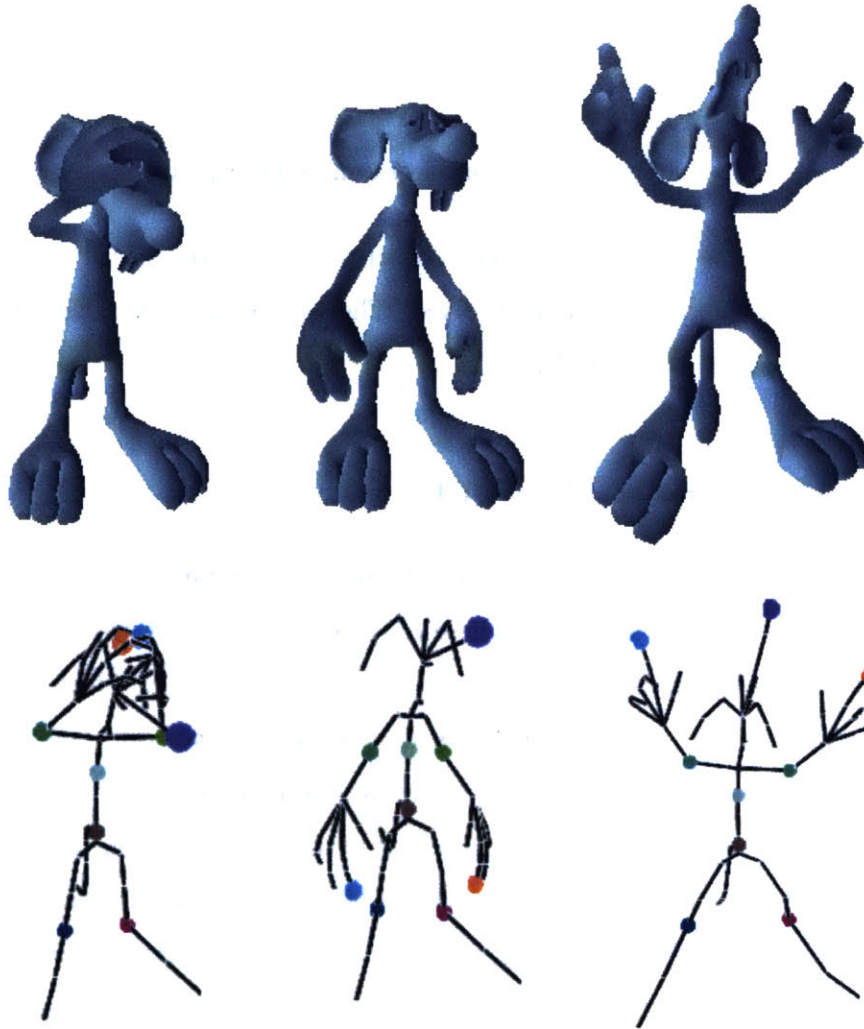


Figure 5-3: This figure shows Morris in 3 poses. The top row is Morris as we see him, while the bottom row is Morris as seen through Max's synthetic vision. The colored spheres on Morris's body are key body parts whose location is tracked by the synthetic vision system.

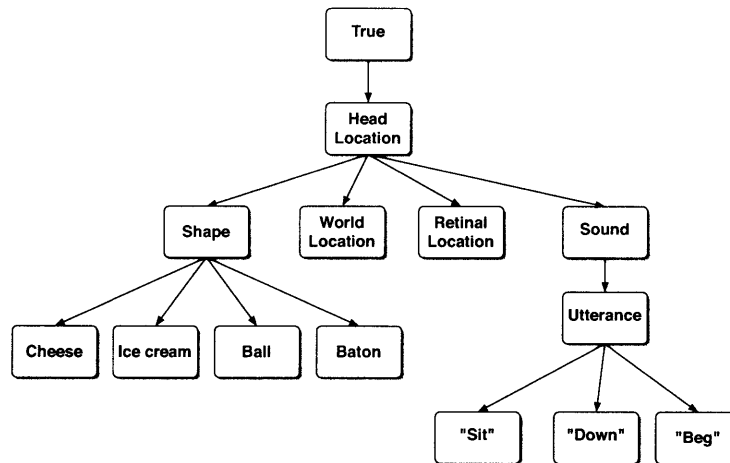


Figure 5-4: A simple example percept tree.

ture’s eye (i.e. the object’s location from the creature’s perspective). See appendix A for the mathematical formulas involved. The specific use of synthetic vision in this thesis is discussed in section 6.1.

5.4 Perception System

Once the *sensory system* has finished filtering the incoming *data records* it passes them into the *perception system*, where they are processed by the *percept tree*. The *percept tree* is a hierarchical mechanism used to extract state and feature information from sensory input. Each node in the tree is called a *percept*, with more specific percepts closer to the leaves. *Percepts* are atomic perception units, with arbitrarily complex logic, whose job is to recognize and extract features from raw sensory data. The simplest of these are *symbolic percepts*, which simply match symbolic input about different objects in the world. For example, a *symbolic shape percept* might simply recognize anything with a “shape”, while its children might recognize the presence of specific shapes, such as a ball, a piece of cheese, or a mouse. Percepts can also perform more complex recognition tasks, such as pose and utterance classification (described in [21]). The root of the tree is the most general percept, which we call *True*. An example percept tree is shown in figure 5-4.

As a *data record* is pushed through the *percept tree*, each *percept* is given the

opportunity to examine the record’s contents. The *percept* then does four things:

- Returns a float between 0 and 1 indicating whether it “matched” the data in the *data record*—that is, whether the data record contained the feature information the percept was looking for. For *symbolic percepts* this match value is generally either 0 or 1 (e.g. the data record is either “ball shaped” or it isn’t), but other percepts, such as classifier percepts or vision based percepts, might return a number in between, depending on how close the input is to their model.
- Returns a “confidence” between 0 and 1, representing how accurate it feels its “match” evaluation is. *Symbolic percepts* almost always have a confidence of 1, since there is no uncertainty in the data they receive. However, percepts using potentially noisy data (e.g. auditory and visual input) might return other confidence values.
- Optionally stores results computed from the incoming data. For instance, a *location percept*, which receives input in the coordinate frame of the character’s eye might choose to store it in other coordinate frames as well, such as relative to the center of the character’s body, or in world coordinates.
- Decides whether to pass the *data record* along to its children for them to evaluate. In general, percepts pass data on to their children when they themselves have matched that data, and don’t pass it on when they haven’t. If an incoming data record has no “shape”, then it certainly won’t have a “ball shape” or a “cheese shape”.

All the data from a particular percept’s evaluation of a data record—match, confidence and any additional results it wants to store, are collected in a time-stamped *percept evaluation*, and all the *percept evaluations* of a particular input data record together form a *belief* about that data record, and are passed into the character’s *belief system*, to be integrated with previous perceptual input, in a process described in section 5.5.

5.5 Belief System

A character’s *belief system* stores its perceptual history, organized into *beliefs* about different objects in the world. A *belief* represents a character’s knowledge about the feature history of a particular object—where it’s been, what it looks like, what it’s said, and so on. *Beliefs* come into the *belief system* from the *perception system* and are first merged with each other, and then with other already existing beliefs.

For example, the same object in the world, let’s say another character, might have a number of data records associated with it. It might put out its own data record containing symbolic information about its shape and location. Meanwhile, the *synthetic vision sensor* might generate a *visual data record* about the location of certain color-coded body parts on this character’s body. These data records go through the percept tree separately, and initially result in two different beliefs. However, since they originate from the same object in the world, the data in these two data records must be combined, and any conflicts in the data between the two (for instance differing position information) resolved.

Beliefs are combined with each other using *merge metrics* which are used to compare all the incoming beliefs to each other. *Merge metrics* simply evaluate whether they believe two beliefs originated from the same object or not. While *merge metrics* can use any mechanism to compare beliefs, the most common methods used are looking at position (“are these beliefs coming from the same place in the world?”) and shape (“do these two beliefs come from objects with the same shape?”). When two beliefs are merged the more reliable of any conflicting data is retained (i.e. the data with the higher confidence values).

Once the incoming beliefs have been merged with each other, they are added to the already existing beliefs in the system. While the incoming beliefs represent what we call a *percept tree snapshot*—the percept evaluations of an object for a particular timestep—the beliefs already in the system represent the character’s knowledge of an object over a period of time. Beliefs contain *percept histories*, which store the match, confidence and associated data returned by each percept, with respect to that object,

per timestep, over a predetermined history length. While it is up to the *percept history* as to how to store this data, the simplest mechanism is to just store all the time-stamped *percept evaluations*. New beliefs are merged into old using the same merge metrics used to combine new beliefs with each other.

5.5.1 Belief Selectors

When other systems, such as the *action system*, want access to information stored in the *belief system* they are able to use a system of *belief selectors*. *Belief selectors* search the belief system for beliefs about objects that fit particular criteria—generally beliefs that match a specified combination of percepts or percept data, for a particular point in time. For instance, a *belief selector* might be used to find a belief about an object that matched *cheese shape* 2 seconds ago. A *belief selector* could also be used to find all the beliefs about objects that matched the *food percept*, and then to select the one among those that’s currently closest to character’s hands.

5.5.2 Derived Percepts

One particularly important kind of percept not described earlier is called a *derived percept*, and operates on a character’s already existing beliefs. *Derived percepts* are able to evaluate objects based on the primitive features other percepts have extracted, and can therefore make more complex assessments of objects, rather than just performing simple feature extraction. For instance, a derived *food percept* could look at the “shape” stored in the *shape percept history* of an object and see that it is “cheese shape”, and then look at the color stored in the *color percept history* and see that it is “yellow”, and match this object as a food object, based on previous knowledge that yellow cheese-shaped objects are food.

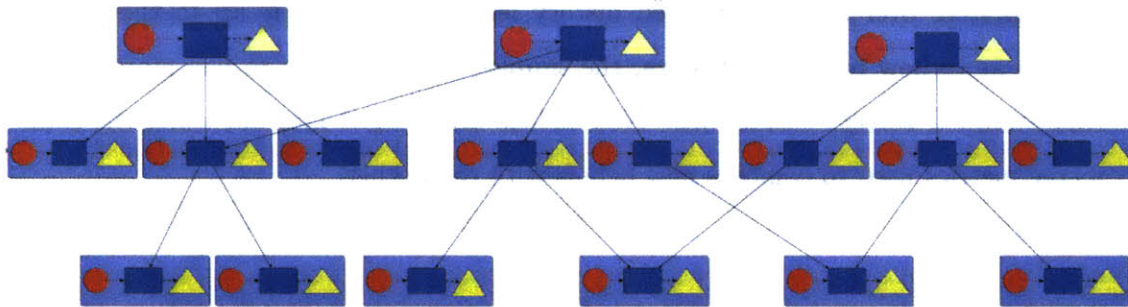


Figure 5-5: An example action system. Purple rectangles represent tuples. Red circles are trigger contexts, yellow triangles are objects, and blue rectangles are actions (do-until contexts not shown). There are three motivational subsystems in this example action system.

5.6 Action System

A character’s *action system* is responsible for behavior arbitration — choosing what behavior the character engages in and when it does so. Individual behaviors are represented in our system as *action tuples* [21] and are organized into a hierarchical structure composed of motivational subsystems (which are described below). An example action system is shown in figure 5-5. Each action tuple contains one or more *actions* to perform, *trigger contexts* in which to perform the action, an optional *object* to perform the action on, and *do-until contexts* indicating when the action has been completed.

The *action* is a piece of code primarily responsible for sending high-level requests for movements or movement sequences to the motor system. The requests can range from something relatively simple such as to “look at” an object, to more complex actions like “reach for the cheese”. Actions in tuples towards the top of the hierarchy are more general (e.g. “satisfy hunger”), and become more specific farther down, with leaves in the action tree corresponding to individual requests to the motor system (e.g. “perform the eating movement”). Actions have associated values, which can be inherent (i.e. pre-programmed) or learned, and represent the utility of performing that action to the creature (for further discussion of action values see [21]).

Trigger contexts are responsible for deciding when the actions should be activated.

In general, there are a variety of internal (e.g., motivations) and external (e.g. perceptions) states that might trigger a particular action. For instance, both the presence of food and the level of a character’s hunger might be triggers for an “eat” action (specific action triggers used in this thesis are discussed in more detail in section 6.2). Similarly, a tuple’s *do-until contexts* decide when the action has completed.

Many behaviors, such as eating and reaching, must be carried out in reference to a particular object in the world. In our system, this object is known as the *object of attention*, and is chosen by a *belief selector* installed in the action tuple. In this thesis, all action tuples not at the top-level of the action hierarchy defer their choice of object to the tuple at the top of their motivational subsystem. Action tuples at the top of motivational hierarchies choose *objects of attention* most likely to satisfy the particular drive they serve (e.g. a *satisfy hunger* tuple might choose a nearby food object), and write these choices into *working memory* for the tuples below them to use.

Action tuples are grouped into *action groups* that are responsible for deciding at each moment which tuple will be executed. Each action group can have a unique action selection scheme, and there can be only one tuple per *action group* active at a time. All the *action groups* in this thesis use a probabilistic action selection mechanism, that chooses among all the tuples they contain based on their respective trigger and action values. As mentioned earlier, the characters in this thesis use an action system that is hierarchically organized and motivationally driven. This hierarchical organization means that each level of the action system has its own action group, containing increasingly specific, mutually exclusive, action tuples. At the top-level are tuples whose purpose is simply to satisfy a particular motivation or drive, such as a *play* or *hunger* drive. Since these tuples are in the same action group, only one of them may be active at a time, which keeps the character from dithering between competing drives.

Below each of these motivational tuples, are tuples representing increasingly specific mechanisms for satisfying drives. For instance, below the *satisfy hunger* action tuple (whose sub-hierarchy is shown in figure 5-6), are tuples such as *get food*, and *eat*

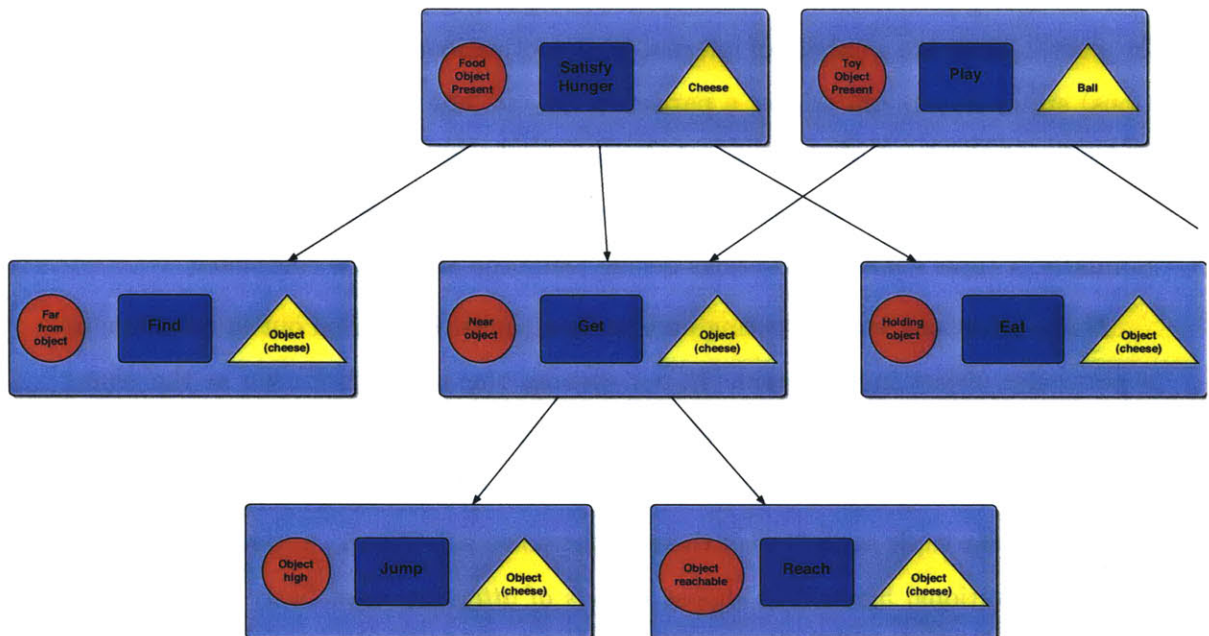


Figure 5-6: A simplified diagram of Max's *hunger* motivational subsystem (the top-level of his *play* motivational subsystem is shown as well)

food, and below *get food* are in turn *reach for food* and *jump for food*. Again, at each of these levels of the hierarchy, only one action tuple at a time may be active. For example, *satisfy hunger*, *get food* and *reach for food* could all be simultaneously active, but *reach for food* and *jump for food* cannot be active at the same time (which makes intuitive sense, since they would require the character to perform conflicting motions simultaneously). I will talk a bit more about drives and motivations in the following section. Finally, one important point about the hierarchical action structure used by the characters in this thesis is its striking similarity to the motivationally-driven hierarchical behavior systems hypothesized by ethologists and cognitive scientists such as Whiten [121], Bryne and Russon [35], and Timberlake [110] [109] (described in section 2.1.1).

5.6.1 Motivations, Drives and Autonomic Variables

Burke [33] provides an excellent description of the use of *autonomic variables* within the Synthetic Characters system:

Our atomic component of internal representation is the Autonomic Variable. Autonomic Variables each produce a continuous scalar-valued quantity. Most Autonomic Variables have drift points values that they drift toward in the absence of any other input. Some of the creature's Autonomic Variables represent Drives, like the hunger drive depicted in the figure below. In addition to its drift point, each Drive also has a set point, the value at which the drive is considered satisfied. The strength of the drive is proportional to the magnitude of the difference between the set point and the output value. Associated with each Drive is a scalar drive multiplier that allows the creature to compare the importance of various drives. Over the course of a creature's existence, these multipliers might change, so that the creature can favor different drives at different times. This mechanism can be used to create periodic changes in the creature's drives (for example, to produce a circadian rhythm) and induce drive-based developmental growth over a creature's lifespan.

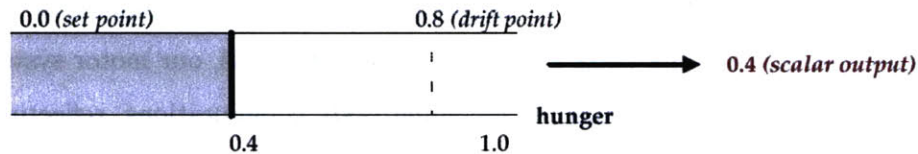


Figure 5-7: An autonomic variable, the atomic component of internal representation (from Burke 2001 [33])

In this thesis, drive values are used as input to both action triggers and action values, so that, for instance, the *satisfy hunger tuple* is triggered by a large rise in the *hunger drive*, while the value of performing the *satisfy hunger tuple* is proportional to the strength of the drive. For more information on autonomic variables and drives, please see the rest of Burke's discussion in [33].

5.7 Motor System

Note: The motor system described in this section is based on the system originally developed by Downie [47], and retains many of its predecessor's representations and mechanisms

For most character architectures, a creature consists broadly of two components—a behavior system and a motor system. Where the behavior system is responsible for working out what the creature ought to be doing, the motor system is responsible for carrying out the behavior systems requests. The primary task of the motor system for a conventional 3D virtual character is therefore to generate a coordinated series of animations that take the character from where his body is now to where the behavior system would like it to be. To do this, our motor system must possess a number of basic competencies: given a set of source animations created by animators, the motor system must be able to play out animations onto character bodies on command e.g. a walk cycle; it must be able to layer animations e.g. a hand wave atop a walk cycle; and it must be able to blend animations—e.g. blending turn left with walk forward to produce an intermediately turning walk cycle.

However, these competencies alone aren't sufficient for accomplishing more sophisticated motor learning tasks such as imitation. For this, our motor system must have additional capabilities, such as modeling body configurations, reflecting on its own contents, coordinating animations with respect to goals (e.g. get my hand close to the food; walk over to the toy and reach for it), and generating novel animations. Additionally, the choice of motor representation becomes critically important if we are interested in the kind of perception-production coupling suggested by Meltzoff's research, research on mirror neurons and Simulation Theory. For this, we need a movement representation that can be used not only to easily generate actions, but to help recognize them. Therefore, because of the importance of motor representation to the goals of this thesis, I will explore the motor system in a bit more detail than I have spent on the systems described so far.

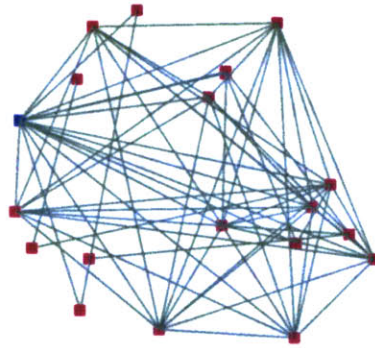


Figure 5-8: A simple posegraph. Green lines are allowable transitions between poses. The Blue square is the pose representing the characters current body configuration.

5.7.1 The Posegraph

Our creatures use multi-resolution, directed, weighted graphs, known as *posegraphs* as their motor representation (An example posegraph is shown in figure 5-8. For a discussion of graph-based motor systems see [47], and also [3] and [73]).

To create a characters posegraph, source animation material is broken up into *Poses* corresponding to key-frames from the animation, and into collections of connected poses known as movements, which generally correspond to individual source animations or motor actions, and are discussed in more detail in section 5.7.3. These representations can be annotated and associated with pre-computed information. Animations can be generated or reformed in real-time by interpolating down a path of connected pose nodes, with edges between nodes representing allowable transitions between poses (or at a lower level of resolution, between movements). This creates both flexibility in the resolution of the motor pieces to be interacted with, and a reduction in the size of the atomic units the motor system operates on. The graph represents the possible motion space of a character, and any motor action the character executes can be represented as a path through the posegraph.

5.7.2 Moving around the graph

The motor system takes the body from an arbitrary pose in the graph to a particular desired pose by searching for the shortest path along the edges of the graph. We use

a popular graph search algorithm known as the A* search algorithm [72]. The use of the A* algorithm will generate search results on demand - rather than a statically computed ‘all pairs shortest path’ algorithm. This allows us to change the distances of edges and change content and topology at run-time without an expensive recomputation.

The motor system travels along the paths found by the A* search algorithm, and generates animations by interpolating between the joint angles contained in the *poses*. If two adjacent nodes originated from the same source animation we already know the time difference between the nodes, because the information is stored in the *pose*. Failing that, we can estimate the time it might take to interpolate between two frames based on the current joint positions and velocities of the two nodes - this calculation is similar to those we perform to find the distance between two poses.

5.7.3 Multi-resolution graphs

As mentioned previously, we can build and store lower resolution views of the motor graph—views of the graph whose nodes are made up of more than one *pose*. These lower-resolution versions of the graph have a number of possible organizations, all of which may be used within a given motor system. Of particular interest, are versions of the graph where pose nodes are grouped into movements, each of which corresponds to an animation (either a source animation or a procedurally generated one).

Movements generally correspond to things we might intuitively think of as complete actions (e.g, sitting, jumping, waving), and therefore often match up closely with requests from the behavior system. While the pose representation provides us with greater motor knowledge and flexibility, the movement representation is often a more natural unit to work with. Our motor system takes advantage of both representations by being able to transition freely between the two views of the graph.

Movements provide the motor system with a shorthand for commonly executed motor actions, allowing comparisons to be made, and information stored, for entire paths through the posegraph. This shorthand speeds up the path discovery process; movements may be used as destination nodes, in which case we need only find the

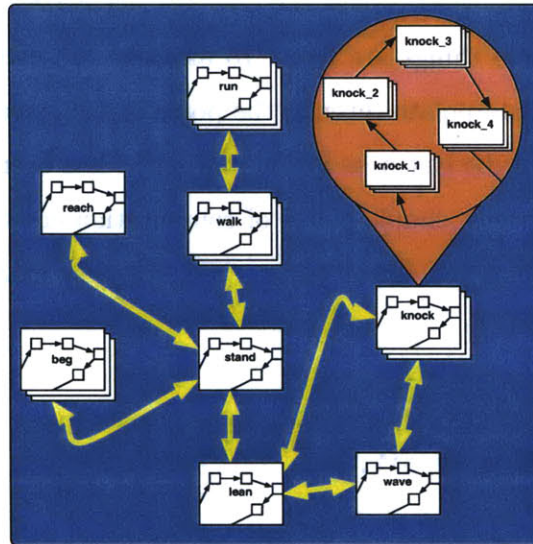


Figure 5-9: An example graph of movement nodes. Large rectangles represent movements, small squares represent poses. Stacks represent blended movements and poses

movement node itself, and then interpolate through the poses it contains, rather than searching each time for the shortest path from pose to pose. In general, lower resolution graph views have far fewer nodes and edges, and can be searched quickly to find areas of the high-resolution graph that may contain the pose that best solves a particular problem. Finally, movements may themselves be grouped, allowing for increasingly coarse views of the graph to be created.

Another multi-resolution aspect of the graph is seen in a representation called the *blended pose*. A *blended pose* is a pose containing sub-poses that it blends together at run-time, in order to derive its joint angle, velocity, timing and other data. We can use this ability to blend between different gaits of walking, different degrees of turning, or for looking in different directions, and to create a continuous output space of poses. .

5.7.4 Motor Programs

There is an important void in the framework described so far— we have no formal way of specifying a complex, coordinated, task (e.g. sit down; turn-left, move forward, move for-ward, stop), or to recognize that a particular task or gesture has been

achieved. In the simplest case, where the complex act corresponds to a previously created movement, these things are free. To animate the complex act, the motor system should simply interpolate through the poses in the corresponding movement; the act is finished when the last pose in the movement has been played out.

We use motor programs, simple pieces of computation created by the motor system, to specify and complete motor tasks. The simplest example program might simply be:

```
GetPath from currentNode to sit;  
Wait until currentNode == sit;  
...
```

Which would tell the motor system to find a path from the node representing the characters current body configuration to the movement node labeled sit, and then to wait for that path to be interpolated and played out on the characters body, before making another motor request. Alternatively:

```
GetPathTo sit start;  
Wait until currentNode == sit start;  
GetPathTo sit middle;  
Wait until currentNode == sit middle  
...
```

would similarly generate a sit animation. Here, the labels sit start and sit middle refer to labeled pose nodes from the original sit animation, rather than movement nodes. Note that while the motor program itself waits for the path it is requesting to complete, the motor system may ask to stop it at any point during its execution, and start another program from the point it leaves off (motor programs in turn can voice opinions as to whether or not they are able to stop). Motor programs responsible for playing out particular motor actions can also let the behavior and motor systems know when the actions have started or completed:

```
GetPathTo sit start;
```

```
Wait until currentNode == sit start;  
Post sit begun  
...
```

Additionally, the computations done by a motor program are not limited to finding paths to predetermined nodes, but can be as far-ranging as searching for the pose that best matches an example body configuration, finding the path that brings a body part closest to a variable location (e.g. get your hands as close as possible to the cheese), and conditionally choosing different destination nodes depending on behavior system (or other) input.

Finally, labels within motor programs can also refer to other motor programs in which case the program:

```
execute sit;  
wait until sit finished;  
execute follow your nose  
...
```

would set a course for whatever node the program sit decides to go to, and wait until the sit program ends, cycles or otherwise allows itself to be interrupted, before starting the follow your nose program, which could itself have many sub-programs. Therefore we build in the ability to have a stack of programs and sub-programs active at any one time. Since these programs can generate destination nodes, and even the contents of destination nodes dynamically, they allow us to generate paths and movements that go beyond recreations of source animation material.

5.8 Summary

This chapter introduced the cognitive architecture used by characters such as Max and Morris. The systems described here allow an animated character (or a robot) to perceive and act upon the world around it, in a natural and life-like way. Furthermore, these systems were designed with the sensory, behavioral, and motor abilities of

humans and animals in mind. In particular, our characters use a hierarchical action system much like that described in section 2.1.1, and our motor representation can be used to implement mirror neuron-like perception-production coupling (we will return to this idea at several points in this thesis). Next, we will explore the specifics behind Max and Morris's behavior.

Chapter 6

Implementation and Results

Now that I have introduced Max and Morris, as well as described their underlying cognitive architecture, it's time to look at the specifics of how the Simulation Theory-style social learning system behind their behavior is implemented.

6.1 Imitation and Movement Recognition

6.1.1 Overview

As described in chapter 4, Max the Mouse is able to observe and imitate his friend Morris's movements, by comparing them to the movements he knows how to perform himself. Max watches Morris through a color-coded synthetic vision system, which uses a graphical camera mounted in Max's head to render the world from Max's perspective (described in section 5.3.1). The color-coding allows Max to visually locate and recognize a number of key body parts (also referred to here as effectors) on Morris's body, such as his hands, nose and feet. Currently, Max is hard-wired to know the correspondence between his own effectors and Morris's (e.g. that his right hand is like Morris's right hand), but previous projects have featured characters using learned correspondences [28], and a similar extension is planned for this research (discussed in section 7.2.1). Similarly, Max starts out knowing which body parts in the image are which (e.g. that yellow is the color-code for the nose, and blue is for the

left hand), which is somewhat analogous to the idea that animals and infants have innate templates for recognizing certain facial features [87].

When Max is asked to watch Morris, he roughly parses Morris's visible behavior into individual movements and gestures. Max locates points in time when Morris was momentarily still, or where he passed through a transitional pose, such as standing, both of which could signal the beginning or end of an action. Max then tries to identify the observed movement, by comparing it to all the movement representations contained within his own movement graph. To do this, Max compares the trajectories of Morris's effectors to the trajectories his own limbs would take while performing a given movement. This process allows Max to come up with the closest matching motion in his repertoire, using as few as seven visible effectors (as of writing, I have not tested the system using fewer than seven). By performing his best matching movement or gesture, Max can imitate Morris. In the following sections, I describe this process in more detail.

6.1.2 Parsing Observed Motion into Gestures

When Max uses his synthetic vision system to watch Morris, he sees an essentially continuous stream of input, broken only into individual frames of graphics. Max thus faces a classic motion capture problem: how to parse data from an ongoing series of actions into individual movements and gestures, and how to recognize these movements and gestures (the problem of recognizing and labeling object motion is introduced by Badler in [6]). One common approach to segmenting motion data is looking for large changes in the acceleration and velocity of key joints—which might represent a body part changing directions or coming to a stop [50] [15]. Often, the 2nd derivative of the motion data is used to detect these points. This information is then combined with a probabilistic model to try and identify whether these points could be the beginnings or endings of movements.

In this case, I took a somewhat different approach to segmenting motion data. Many different movements start and end in the same transitional poses, such as standing or sitting, so that these poses can potentially be used as segment markers.

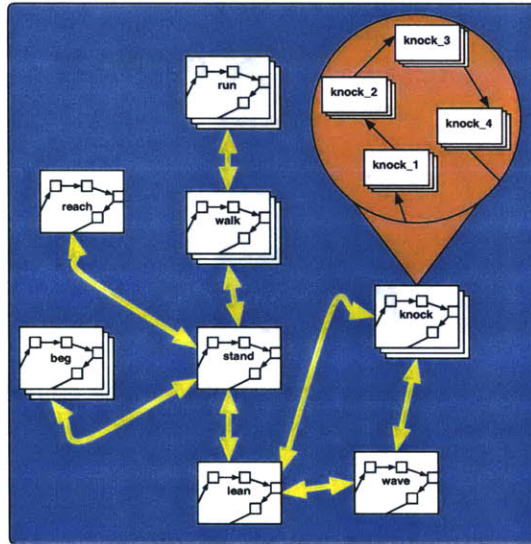


Figure 6-1: An example graph of movement nodes. Large rectangles represent movements, small squares represent poses. Stacks represent blended movements and poses (this figure is the same as figure 5-9)

In fact the idea that certain poses or animations represent “hubs” in a character’s movement repertoire has previously been used in assembling motor graphs for animated characters, and even for parsing human motion capture data [56]. As the example movement graph in figure 6-1 shows, our characters’ movement repertoires generally follow just this sort of hub and spoke model. Here, I have taken advantage of the fact that almost all of Max and Morris’s movement primitives—that is, all the animator-provided source animations that have been assembled into their pose and movement graphs (as described in section 5.7)— pass through a standing position. Rather than finding the 2nd derivative of the motion data for each effector, I use the simpler approach of using places where Morris passes through a hub (in this case standing), as potential indicators of the beginnings and endings of movements.

Looking for Movement Hubs—An Example

Let’s say that Max watches Morris first jump up in the air, and then cover his face with his hands. How does he take this continuous image sequence and correctly divide it into two gestures (rather than one or four or ten)? As described in section

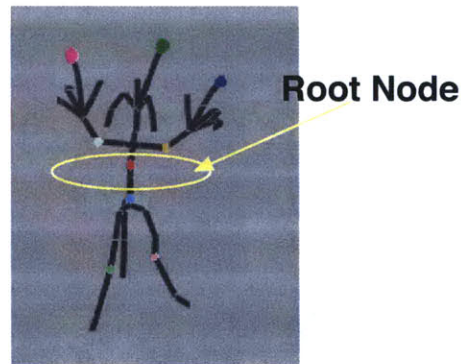


Figure 6-2: Morris viewed from Max's perspective. The colored sphere marking his root node is circled in yellow.

5.3.1, when Max watches Morris, his synthetic vision sensor extracts the world-space position of each of Morris's visible body parts (the use of body part positions and the choice of which parts to use is discussed in the following section). The most important of these positions is Morris's root node (see figure 6-2), which Max uses as a reference point for Morris's movements—converting the world-space position of Morris's other body parts to root node relative positions (e.g. where Morris's hands, elbows and feet are relative to the center of his body, rather than to the center of the world). In order to do this, Max must figure out which way Morris is facing. For this, he uses another visible body part—Morris's nose, as a forward reference. Max assumes that Morris's nose is always in front of the center of Morris's body, and uses the vector between the two points as Morris's forward vector. These pieces of information—the world-space position of Morris's root node and the direction he is facing—are sufficient for a standard coordinate frame transformation.

Max also has another important source of information—himself. By looking at his own *stand* movement, Max can find where his body parts are positioned relative to his root node while he is standing, forming an example standing pose. Max can then compare this example pose with each incoming frame of motion data, to see how close Morris's current position is to standing. The distance metric used to do this comparison is extremely simple:

$$\text{dist}(A, B) = \frac{\sum_{i=1}^N d(A_i, B_i)}{N} \quad (6.1)$$

Where N is the number of visible body parts, A is Max's sample standing pose, B is the observed pose, and A_i and B_i are the x, y, z coordinates of body part i within those poses. The distance between A_i and B_i is given by d , defined as:

$$d(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2} \quad (6.2)$$

where a and b are 3-dimensional vectors. In other words, the distance between standing and an incoming pose is the average distance between the body parts in both poses. For each pose, only the currently visible body parts are used in the comparison (i.e. if the hands are currently obscured, they are left out of the distance metric), which is why an average is used.

Going back to our example, this means that as Morris begins to jump, the distance between his current position and standing increases. At a certain point, an empirically determined threshold is reached, indicating that he is no longer standing (see figure 6-3). Conversely, as Morris falls back to the ground, and starts returning to a standing position, the distance between his current pose and standing drops, until it is once again below threshold. As Morris raises his arms to cover his eyes, the distance of his current pose from standing begins to increase again, and the threshold distance is crossed once more.

In other words, simply by keeping track of when the threshold between standing and not-standing is crossed, and whether it was crossed on a rising or falling edge, Max can find the beginnings and endings of Morris's movements. Added accuracy can be obtained by low-pass filtering the distance values, but as we will see in the next section, only a roughly accurate parsing of the motion data into individual movement segments is necessary in order for Max to correctly identify the movements he sees Morris performing. A nice aspect of this approach is that Max can use his body-knowledge—the knowledge that his movements tend to start and end in hubs, and

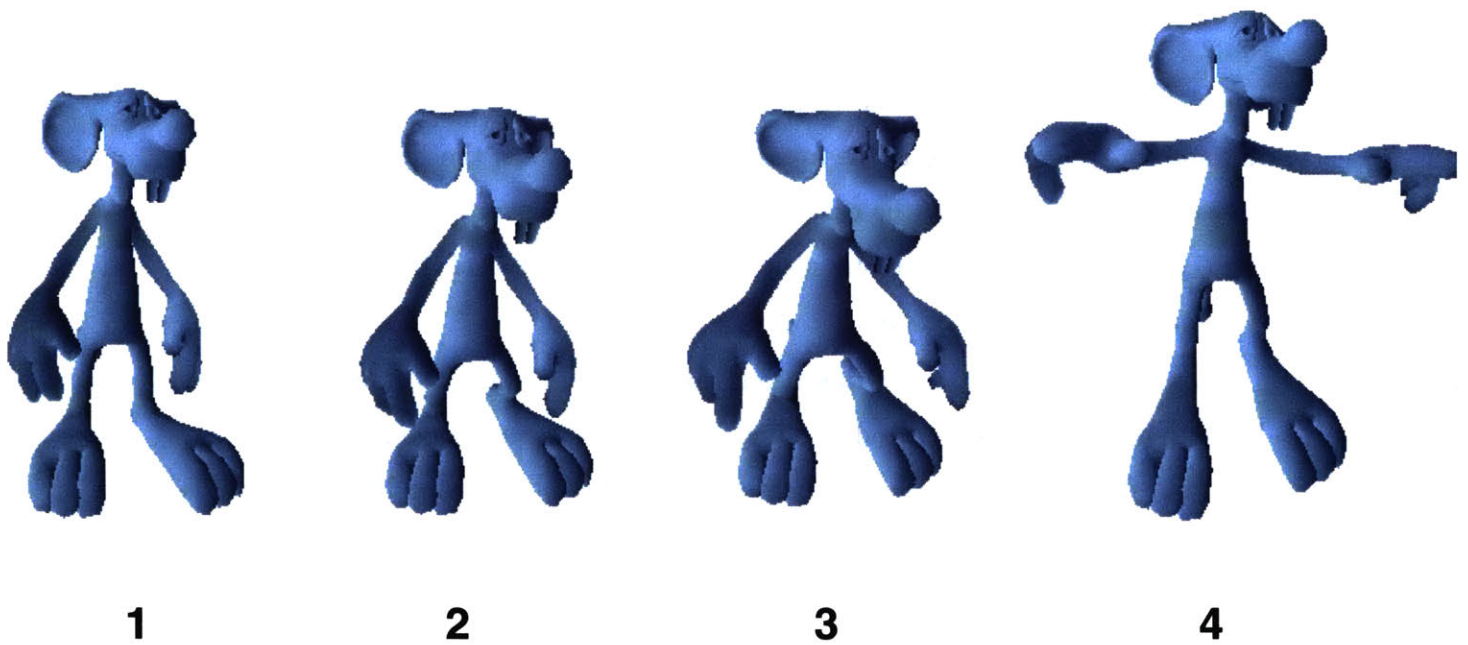


Figure 6-3: This figure shows 4 frames of Morris moving into a jump. In frame 1 Morris starts in a standing position. by frame 3 his body position has changed enough to cross the threshold between standing and moving.

the knowledge of what his own hubs (standing) look like—to simplify the motion parsing problem.

Other Approaches to Movement Parsing

While the “hub-and-spoke” approach to movement parsing presented here turned out to be sufficient for our purposes, we have also implemented some supplementary movement parsing mechanisms. These additional mechanisms can provide help at points of ambiguity, where it is unclear if a transition has occurred or not, and could potentially be used to parse more challenging data sources, such as human motion capture data.

The first such mechanism is a system that models the demonstrator’s movements over time, and identifies points of return (i.e. particular positions or poses within the motion that the demonstrator keeps returning to). These locations often represent significant breaking points in the motion, particularly in cyclical movements such as reaching, walking, jumping etc.

Additionally, we have implemented a mechanism that keeps track of when key effectors (e.g. the hands and feet), come close to, or draw away from, objects in the environment, which again, might indicate the beginning or ending of a movement (for further discussion of the limits and extensibility of our movement parsing approach, see section 7.2.2).

Using Visible Body Part Locations

Frequently in movement recognition research, input data is received in the form of joint angles gathered from motion capture suits. I chose to use the positions of key body parts rather than joint angles as my input for reasons of biological plausibility. First, while joint angle data is easily available from motion capture suits (and currently, is often the only way to gather human data) and potentially from synthetic environments, it is much more difficult to calculate from visual information alone. It seems unlikely that humans and other animals are making extensive joint angle inferences during their observations of others, and similarly, calculating joint angle

positions from synthetic visual data is much more difficult than simply finding body part positions. Furthermore, research has shown that people who are watching others move spend most of their time attending to the locations of certain key body parts, like hands and elbows [50]. I attempted to choose equivalently salient body parts—the hands, elbows, knees, feet, torso (root node), and nose. Additionally, effector positions turn out to be a very flexible form of input. The system has been successfully run using only the hands, feet, torso, nose and neck, and, as described below, compensates well for partial or obscured data.

6.1.3 Matching Observed Gestures to Movements in the Graph

Once Max has seen Morris perform a complete action, he is faced with another classic problem: movement recognition. Max must classify the movement he has seen as one of a set of movements (in this case the set comprised of the movements he is able to perform himself). There have been many computational approaches to the problem of gesture and movement recognition from visual data (see for example [125], [16], [23] and reviews in [54] and [127]), most of which use a set of probabilistic models to classify gestures, and rely on a large pre-existing example set. Here, rather than providing Max with a large set of example movements, we note that he *already* has a built-in example set—his own movement repertoire, represented by his posegraph.

As Max watches Morris demonstrate a gesture, he represents each frame of observed motion by noting the world-space positions of Morris’s body parts relative to the center of Morris’s body. He then searches his posegraph for the poses (frames) closest to the beginning of the observed action (e.g. poses with similar hand, nose, and foot positions to those he’s seen). Max uses these best-matching poses as starting places for searching his posegraph, exploring outward along the edges from these nodes, and discarding paths whose distance from the demonstrated gesture has become too high. Max can then look at the path generated through his graph and see whether it corresponds closely to any of his existing movements, or whether it represents a novel gesture. In the next section, I will describe movement matching in a bit more detail.

Matching Observed Movements—An Example

Let's go back to the example of Max watching Morris jump in the air and then cover his face with his hands. How does Max identify these gestures with his own *cover face* and *jump* movements?

To briefly review from section 5.7, Max's motor representation consists of a multi-resolution graph. At the lowest resolution, nodes in the graph represent individual frames of animation, while at a higher level, nodes represent motion primitives (complete animations) called *movements*, which are collections of connected poses, forming a path through the graph. Edges in the graph represent allowable transitions between nodes, and animations are formed by interpolating along paths in the posegraph.

When Max sees Morris jump, he represents jumping as a sort of path as well—as a sequential series of poses containing the root-relative positions of Morris's body parts. So, in order to represent Morris's *jump* in his own motion space, Max needs to find the path through his posegraph that is 'closest' to the path he observed. Max chooses a frame from the middle of the observed jump as a starting point for identifying what he's seen. The middle of a movement tends to be more representative (i.e. less generic or similar to other movements) than the beginning or the end. He then searches all the poses in his graph for those most similar to this representative frame, using the distance metric defined in equation 6.1.2. Max next searches outward from each of these best-matching poses, trying to assemble the overall best-matching path, and pruning his search as he goes along.

The following set of figures walks through a very simple example of the matching process. In this first figure, the bright green circle in the observed movement is the first pose to be matched, and the bright green circles in the posegraph are the two best matches:

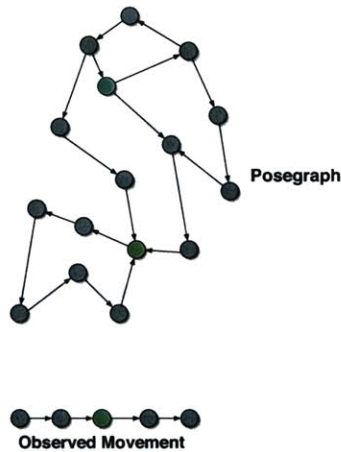


Figure 6-4: The movement matching process—step 1

Now the next frame in the observed movement is compared to the children of the initial matches (the children are shown in blue): This gives us four potential paths

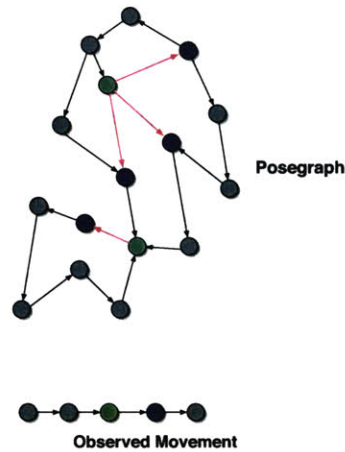


Figure 6-5: the movement matching process—step 2

(since there are four potential next poses). Only the best-matching of these paths will be kept, where the distance between the observed movement O and a given path P is defined as:

$$pathDistance = \sum_{i=1}^N dist(O_i, P_i) \quad (6.3)$$

Where N is the length of the path currently being considered (in this case, two poses long).

Let's say that, in this example, we keep only the two best paths each time:

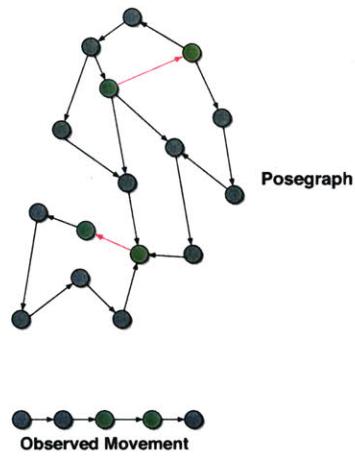


Figure 6-6: The movement matching process—step 3

Next, we work in the other direction, looking at the parents of each potential path (shown in red):

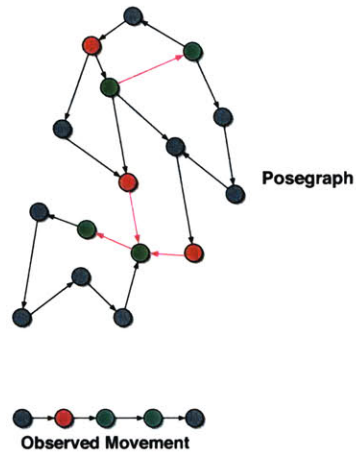


Figure 6-7: The movement matching process—step 4

Again, we keep only the two best-matching paths:

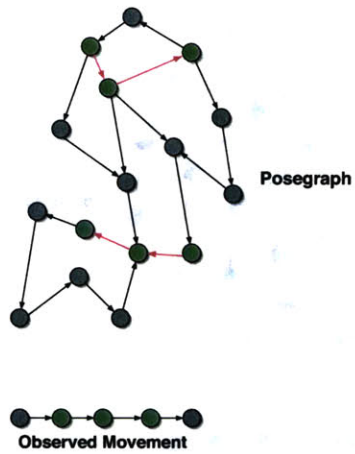


Figure 6-8: The movement matching process—step 5

The process is repeated for the remaining poses in the observed movement:

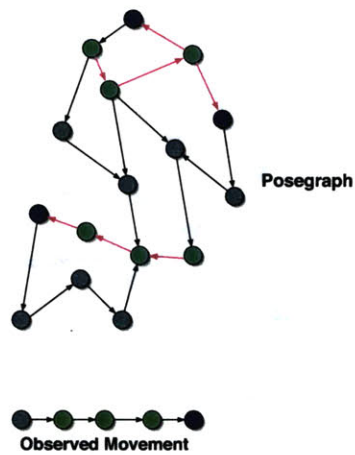


Figure 6-9: The movement matching process—step 6

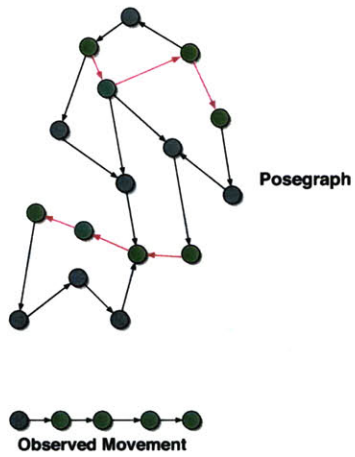


Figure 6-10: The movement matching process—step 7

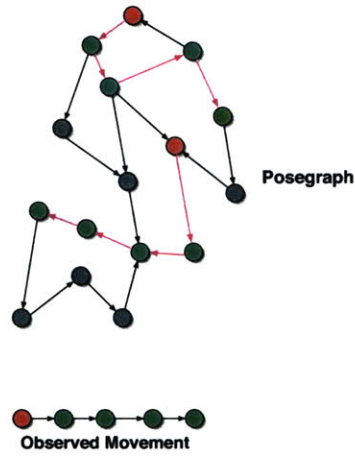


Figure 6-11: The movement matching process—step 8

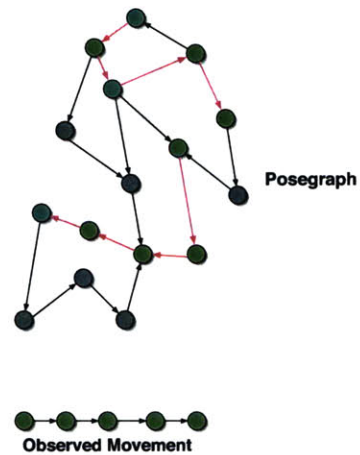


Figure 6-12: The movement matching process—step 9

Finally, we choose the best-matching of the two remaining paths:

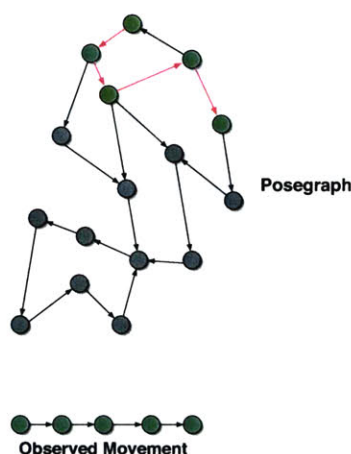


Figure 6-13: The movement matching process—step 10

Max then takes this best-matching path and checks to see which movements the poses in the path are part of. If the majority are from a particular movement (in this case, jumping), Max assumes that this is the movement he saw Morris performing.

Distinguishing Similar Movements

Often, characters must differentiate between two very similar looking movements (or combinations of movements). Max and Morris have a number of distinct gestures that use the same body parts and produce similar motion trajectories, such as waving vs. knocking vs giving a thumbs up, reaching high up vs jumping, jumping to reach something vs jumping for joy, and quite a few others. While we have not set out to explicitly test the limits of our system’s ability to distinguish between similar movement primitives, the experience in implementing this thesis has been that movements that are similar, but visibly different to human observers, produce enough subtle variation in the body position, trajectory, or speed of a movement to be successfully distinguished by our system (whether this would remain true with noisier vision data is harder to say—see section 7.2.2 for a discussion).

Characters With Differing Posegraphs

Until now, we have also only been discussing characters who share the same movement space (i.e. they have identical posegraphs). What happens to the movement recognition process if Max and Morris don't have the same movement primitives?

We created versions of Max and Morris with somewhat different posegraphs, where a subset of each character's movements were similar, but not identical, to a corresponding set of the other character's movements. Again, while we did not explicitly test the limits of the system, our observation is that, in these situations, Max picked what appeared to be the closest approximation of the gestures he saw Morris performing.

For example, Max had a movement in his motor system that involved covering his face with his hands. Meanwhile Morris had a similar movement, where he covered his face and shook his head, rocking back and forth. Morris's movement took longer than Max's to perform, and involved the motion of a number of additional body parts. Nevertheless, Max identified his own, non-identical, *cover face* movement as the closest match in his system.

There is a relatively graceful degradation to the matching process between non-identical graphs. As the movement primitives the two characters possess become more disparate, the closest match becomes coarser (e.g. when Max has no *cover face* equivalent at all, the best match to seeing Morris cover his face is to reach up with his arms near his head). Ultimately, the character might decide that what he is seeing doesn't match any of his existing movements at all, and is instead a completely novel movement (see section 6.1.3).

Blended Movements

Another way in which differences in movement repertoire are dealt with in this implementation is through the use of blended movements (described in section 5.7.3). For example, Max's *reach* movement is a blended movement, composed of nine separate reaching animations, which, when blended together, allow him to reach all around his

body. By comparing Morris's movements to each of the sub-movements in *reach*, Max can see if the movement he observed is contained within the space the sub-movements define, even if it doesn't correspond directly to any of them.

Identifying Novel Movements

In the case where Max and Morris have posegraphs containing different movements, Morris could perform a movement that doesn't closely match any of the movements in Max's repertoire. Max decides that a movement he has observed is novel when the best-matching path through his graph doesn't closely correspond to any of his existing movements (i.e. the poses in the path are contained within many different movements, or aren't traversed in the order any existing movements traverse them). The use of novel movements is discussed briefly in the next section, and in more detail in section 7.2.6.

Advantages of Graph-based matching

One important benefit of using the posegraph to classify observed motion is that it simplifies the problem of dealing with partially observed (or poorly parsed) input. If Max watches Morris *jump*, but doesn't see the first part of the motion, he will still be able to classify the movement as *jumping* because the majority of the matching path in his posegraph will be contained within his own *jump* movement. Conversely, if Max has observed a bit of what Morris was doing before and after *jumping*, as well as the *jump* itself, he can use the fact that the entire *jump* movement was contained within the matching path in his graph to infer that this is the important portion of the observed motion. In general, this graph-based matching process allows observed behaviors to be classified amongst a character's own actions in real-time without needing any previous examples.

Additionally, while this functionality has not yet been taken advantage of, a graph-based matching system makes it easy for a character to learn completely novel movement primitives through observation. If the matching path in the graph does not correspond to any existing *movements* it can be grouped into a new *movement*, since



Figure 6-14: Morris covering his eyes, as seen by Max. Notice that some of the spheres marking his body parts are not visible. This is a repeat of figure 4-9.

a movement is just a path through the posegraph.

Finally, another important note is that the combination of using effector locations as input, and using a graph-based matching process, appears to compensate well for the natural obstructions of visibility and changes in viewpoint that occur when one creature is observing another. For example, Max is able to correctly identify and imitate Morris's *cover face* movement, even though his nose and hands are not visible at several points during the movement (see figure 4-9).

6.1.4 Imitation

Once Max has seen Morris jump in the air, and identified this movement as jumping, Max's action system can request a jump movement from his motor system, allowing him to imitate Morris. One important aspect of this implementation of imitation is that it uses a Simulation Theory-style approach in order to give one character knowledge of what the other has done. In particular, not only is Max's own motor representation used to classify Morris's movements, this classification is done explicitly—that is, Max doesn't just play out the animation generated by the matching path in his posegraph, he performs the movement this path most likely represents. This means that when Max sees Morris jump, he can not only imitate that jump, but identify it with his own jumping, and begin to look at what jumping is often used for. Coupling the perception (classification) and production of movements allows Max to begin ex-

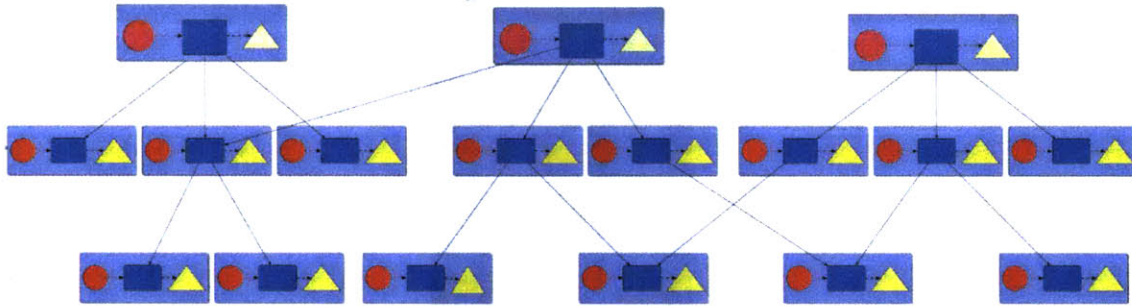


Figure 6-15: An example action system. Purple rectangles represent tuples. Red circles are trigger contexts, yellow triangles are objects, and blue rectangles are actions (do-until contexts not shown). This figure is a repeat of figure 5-5.

aming the motivations and goals for these movements, an idea I will explore in the next few sections.

6.2 Identifying Actions, Motivations and Goals

We just saw that, by matching observed gestures and movements to his own, Max is able to imitate Morris. Max can also use this same ability to try and identify which actions he believes Morris is currently performing.

As described in section 5.6, Max and Morris both choose their actions using motivationally driven, hierarchically organized action systems, composed of individual action units known as *action tuples* (detailed in [21]). Each action tuple contains one or more *actions* to perform, *trigger contexts* in which to perform the action, an optional *object* to perform the action on, and *do-until contexts* indicating when the action has been completed. Within each level of the action hierarchy, tuples compete probabilistically for expression, based on their action and trigger values. Action tuples towards the top of the hierarchy are more general (e.g. satisfy hunger), and become more specific farther down, with leaves in the action tree corresponding to individual requests to the motor system (e.g. perform the eating movement).

Max keeps a record of movement-action correspondences, that is, which action he is generally trying to carry out when he performs a particular movement (e.g. the

reaching gesture is most often performed during the *getting* action). When he sees Morris perform a given movement, he identifies the action tuples it is most likely to be a part of. He then evaluates a subset of the trigger contexts, known as *can-I triggers*, to determine which of these actions was possible under the current circumstances. In this way, Max uses his own action selection and movement generation mechanisms to identify the action that Morris is currently performing. The following sections describe this process in greater depth.

6.2.1 Action Identification: Example 1

Let's say that Max sees Morris eating a piece of cheese. How does Max identify that action as eating, and how does he know that it is part of the *hunger* motivational subsystem (shown in figure 6-16)?

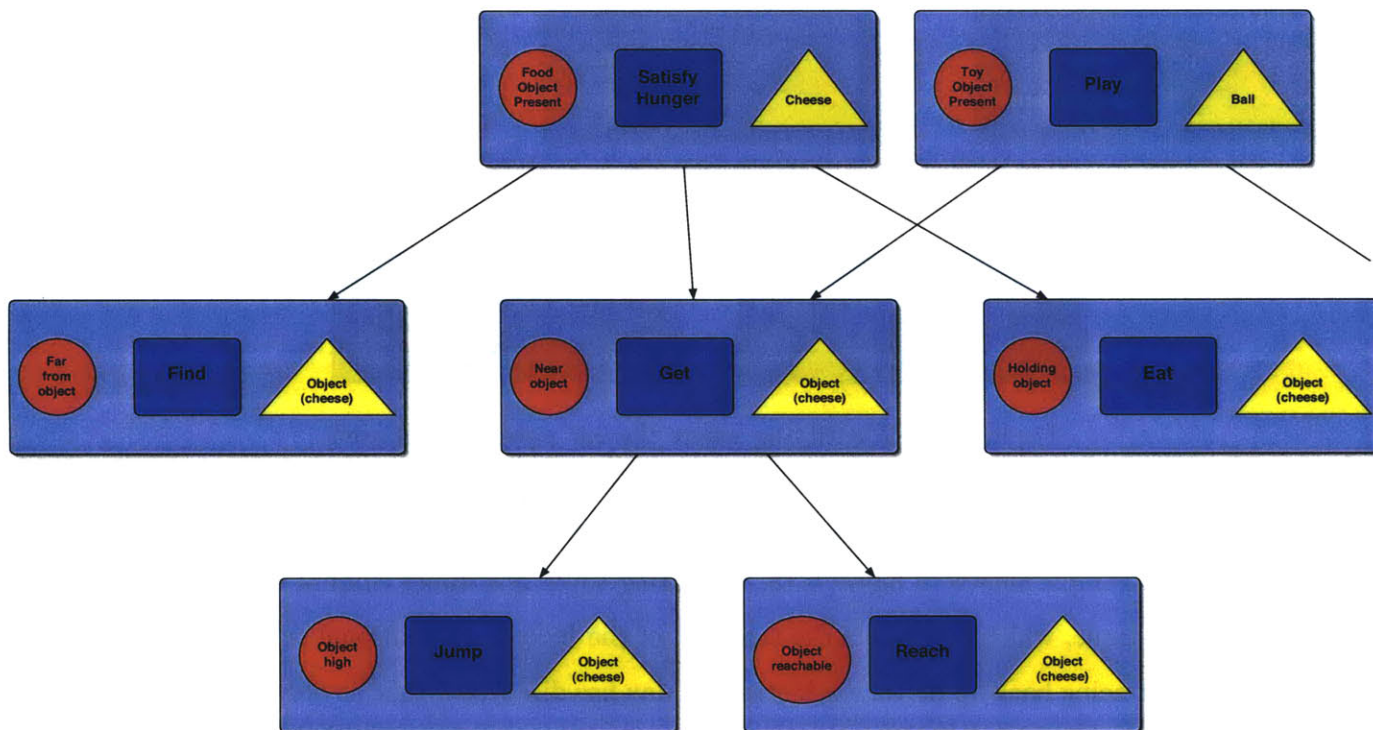


Figure 6-16: A simplified diagram of Max's *hunger* motivational subsystem (the top-level of his *play* motivational subsystem is shown as well). This figure is the same as figure 6-16 seen earlier.

When Max sees Morris eat, the first thing he does is identify the movement he

sees Morris performing (as described in section 6.1.3). In this case that movement is *eating*. Next, Max searches his map of movement-action correspondences to find out which of the action tuples in his action hierarchy have requested this movement in the past. Recall that a movement is an individual motion primitive, such as *reaching*, *jumping* or *eating*, while an action is a behavior that occurs in a motivational and environmental context, and requests that the motor system carry out particular movements. In this case, Max finds that he has only performed the *eating* movement during his *eat* action (figure 6-17).

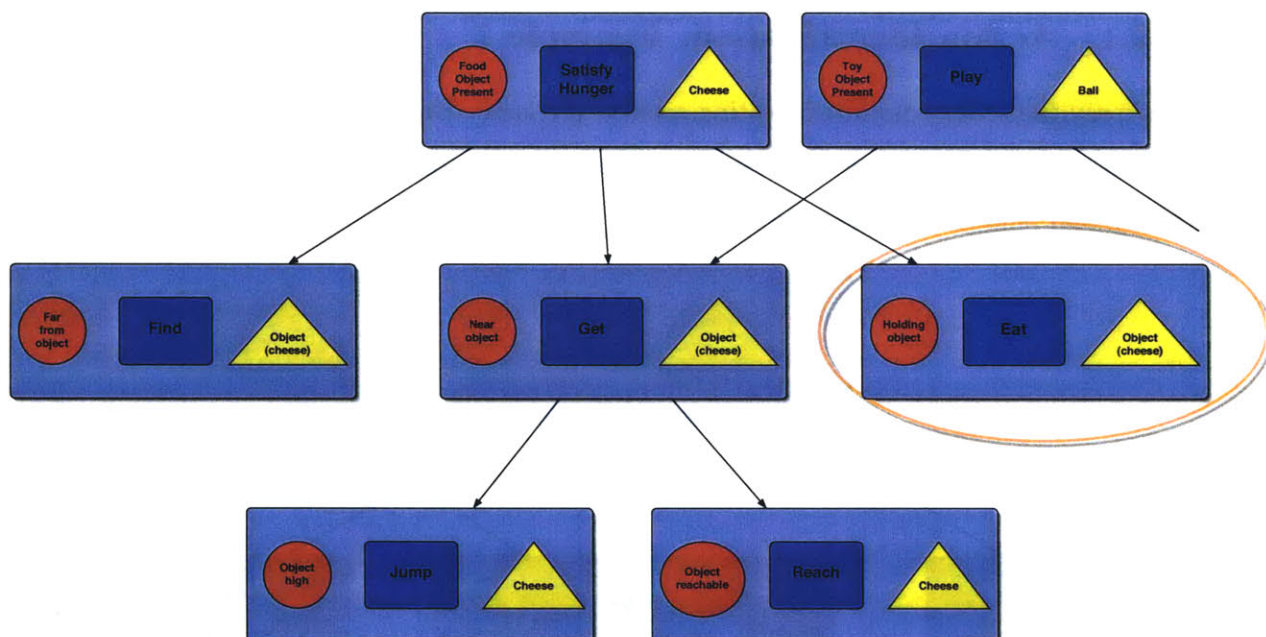


Figure 6-17: Identifying where the *eat* movement is used

Max then traces back up his action hierarchy from the eating action tuple. In the simple example shown in figure 6-18, the eating tuple is a direct child of the *satisfy hunger* tuple, which represents the top level of the *hunger* motivational subsystem.

By tracing back up his action hierarchy, Max has discovered that he only performs the *eating* movement when his *eat* action tuple is active, and he only uses his *eat* action tuple when he's trying to satisfy his hunger. Therefore, Max now knows that it's likely that Morris was eating, and that he was eating because he was hungry. Now, Max must verify that it was possible for Morris to be eating, given the environmental

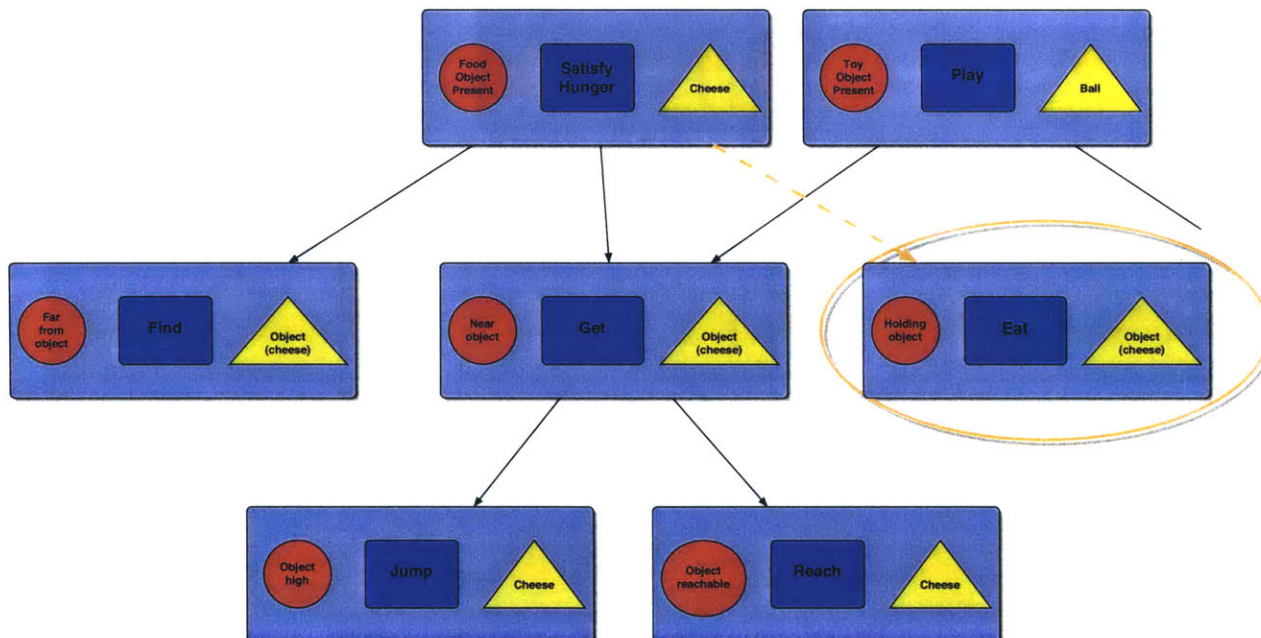


Figure 6-18: the path through Max's action system to his eating action

circumstances. To do this, Max once again uses a simulation theoretic approach—he checks whether it would have been possible for him to eat if he were in Morris's place.

Max evaluates the *can-I* triggers at each level of his action hierarchy, on the way to the *eat* action tuple. In this case, there are only two levels, with one *can-I* trigger each (figure 6-19). At the top-level, the *can-I* trigger *food object present* simply checks whether there are any food objects in the environment. This kind of trigger is known as an *object selection trigger*, since it checks whether any appropriate objects are available, and then selects one of them as the *object of attention* for all the actions below it in the hierarchy. The simplest *object selection trigger* simply searches the *belief system* for *beliefs* about objects that have certain perceptual features. To review from section 5.5, the *belief system* stores a character's representation of perceptual input in the form of *beliefs*, which generally correspond to individual objects in the world, and contain *percept histories* of what the object's features have been over a short time period (e.g. the object's shape, color, location etc.). In this case, Max has a *derived percept* that recognizes food items, and so the *food object present* trigger looks for *beliefs* about objects that the *food percept* has fired on (i.e. objects that have

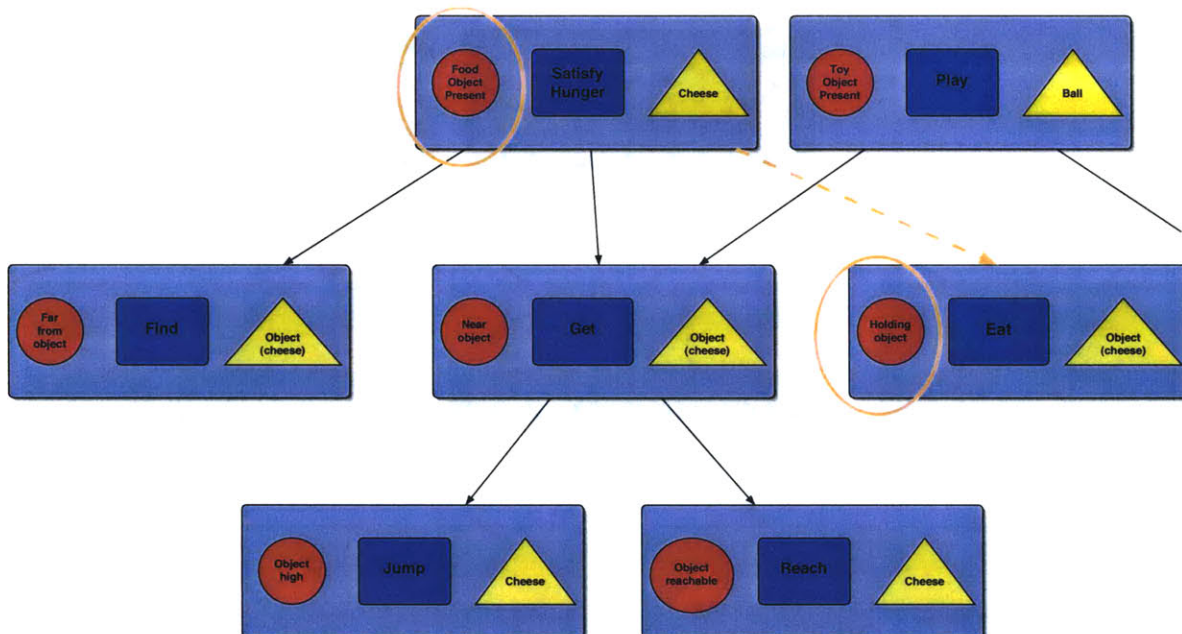


Figure 6-19: Evaluating the *can-I triggers* along the path to Max's eating action

been marked as food). For more information on the belief and perception systems please see chapter 5.

One subtle but important point here is that Max must check whether there was any food **at the time he saw Morris eating**, not whether there is any food available currently, which is what he would check if he himself were hungry. Luckily, since the *belief system* stores *percept histories* for each *belief*, Max's *food object present* trigger can simply search for food objects at the time Morris began eating, rather than at the current time. In this case, the *food object present* trigger finds that a piece of cheese was available, and sets the *belief* about this piece of cheese as the *object of attention*.

Now there is only one more *can-I* trigger to check. Max's *eat* action tuple can only be activated if he is currently holding a piece of food (see figure 6-19). Max's *holding food* trigger is a *proximity trigger*, which checks if two objects are within a certain distance of each other. Normally, the *holding food* trigger checks whether the *object of attention* is in the same place as one of Max's hands. However, because he is currently simulating Morris's situation, in this case the *holding food* trigger must instead check if the *object of attention* was in the same place as one of **Morris's**

hands. Since Morris's hands are both visible body parts, their positions when he started eating are stored in Max's *belief* about Morris. In other words, Max can simply evaluate his *holding food* trigger using his belief about Morris instead of his belief about himself (representations of self and other are described in more detail in section 6.2.4). Here, he finds that Morris was in fact holding a piece of cheese at the beginning of the *eating* movement. The *holding food* trigger returns true, and Max concludes that Morris was indeed eating because he was hungry, and demonstrates this by miming eating.

In general, evaluating Max's *can-I* triggers from Morris's perspective is almost identical to evaluating them from Max's perspective, with just two important changes:

- Evaluation occurs for the timestep in which the observed action began, not for the current timestep
- Max's *belief* about Morris is used everywhere he would normally use his *belief* about himself

In the following section, I will go through an example where Max uses this technique to identify a more ambiguous action.

6.2.2 Action Identification: Example 2

Let's say that Max sees Morris reaching for a piece of cheese instead of eating one. Once again, how does he identify what Morris is doing?

As before, Max first identifies the movement he saw Morris performing, in this case *reaching*. When Max looks in his action-movement correspondence map, he finds that the *reaching* movement is used by his *reach* action-tuple. By tracing up his action hierarchy from the *reach* tuple, Max finds that *reach* is part of the *get* tuple, which is used by a number of motivational subsystems. In the example shown in figure 6-20, *get* is used by both the *hunger* and *play* motivational subsystems.

Since Max uses the *reach* action tuple in a number of contexts, he must decide which one of these contexts best matches Morris's current situation, in order to decide *what* Morris is reaching for, and *why* he is reaching for it.

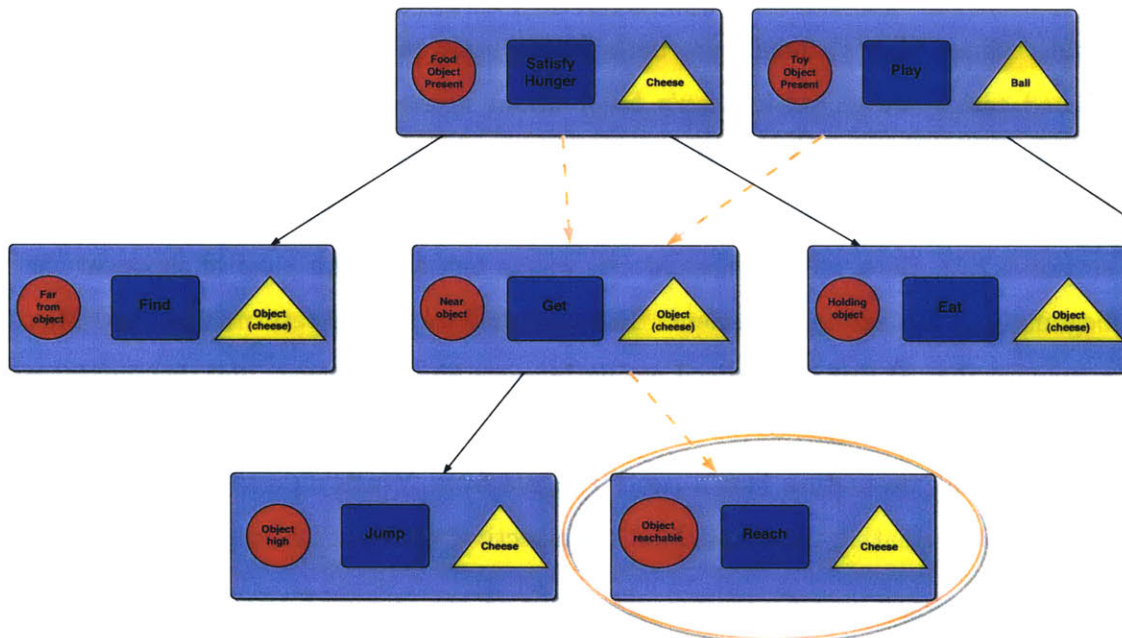


Figure 6-20: The paths through Max's action hierarchy to the *reach* action. Notice that it can be activated in both the *hunger* and *play* motivational subsystems

Max's movement-action correspondence map includes not only bottom-level action tuples such as *eat* and *reach*, but also top-level ones such as *satisfy hunger* and *play*. That is, when Max reaches while his *satisfy hunger* tuple is active, he remembers that he performed the *reaching* movement during the *reach*, *get*, and *satisfy hunger* tuples (see figure 6-21). In fact, the movement-action correspondence map is simply a list of the actions that have been active when a particular movement was performed, and the number of times that action was active for that movement. Therefore, Max knows which motivation he is most often trying to satisfy while *reaching*—whichever top-level tuple has the highest count for the *reach* action.

Let's say that in this case, Max has performed the *reach* action in order to play more often than he has used it to satisfy hunger. He will start out by guessing that Morris was reaching in order to play, since this is usually why he himself reaches, and will then evaluate his *can-I* triggers along the path from *play* to *reach* to see if he's correct (figure 6-22).

The first *can-I* trigger in the *play* motivational subsystem *toy object present* is

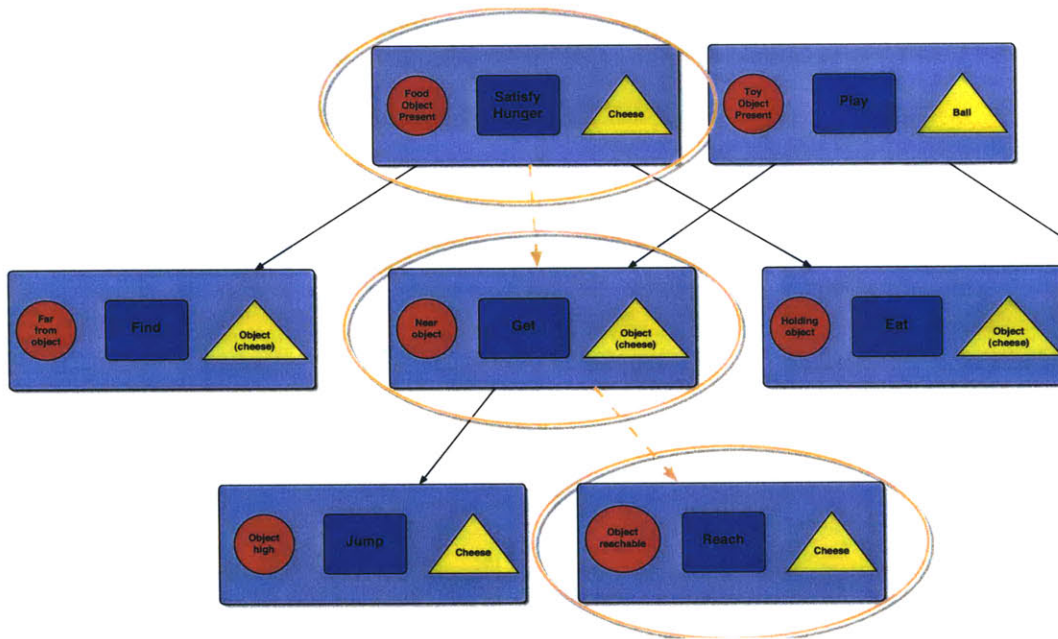


Figure 6-21: When Max reaches for a piece of cheese in order to satisfy his hunger, the *satisfy hunger*, *get* and *reach* action tuples are marked as having been active during the *reaching* movement in Max’s movement-action correspondence map

essentially identical to the *food object present* trigger discussed in the previous section, except that it checks for objects labeled as toys rather than those labeled as food. If there weren’t any toys available, Max’s evaluation of this path through the action hierarchy will stop right here—he’ll know that Morris couldn’t have been reaching for something to play with, because there were no toys to reach for.

Let’s say that, when Max saw Morris reach, there was a ball available, as well as a piece of cheese, but that the ball was high up in the air. In that case, the *toy object present* trigger will return true, and Max will continue evaluating this path through his action hierarchy. The next *can-I* trigger is a *proximity* trigger for the *get* action, which checks whether the *object of attention*—in this case the ball—is close enough to ‘get’ (but not so close that the character is already holding it). If the ball is close enough to Morris that he could have gotten it by reaching or jumping (without needing to walk anywhere), then this trigger will also return true.

Now, there is only one *can-I* trigger remaining—the *object reachable* trigger for the *reach* tuple. Since the ball is high in the air, where Morris would have needed to

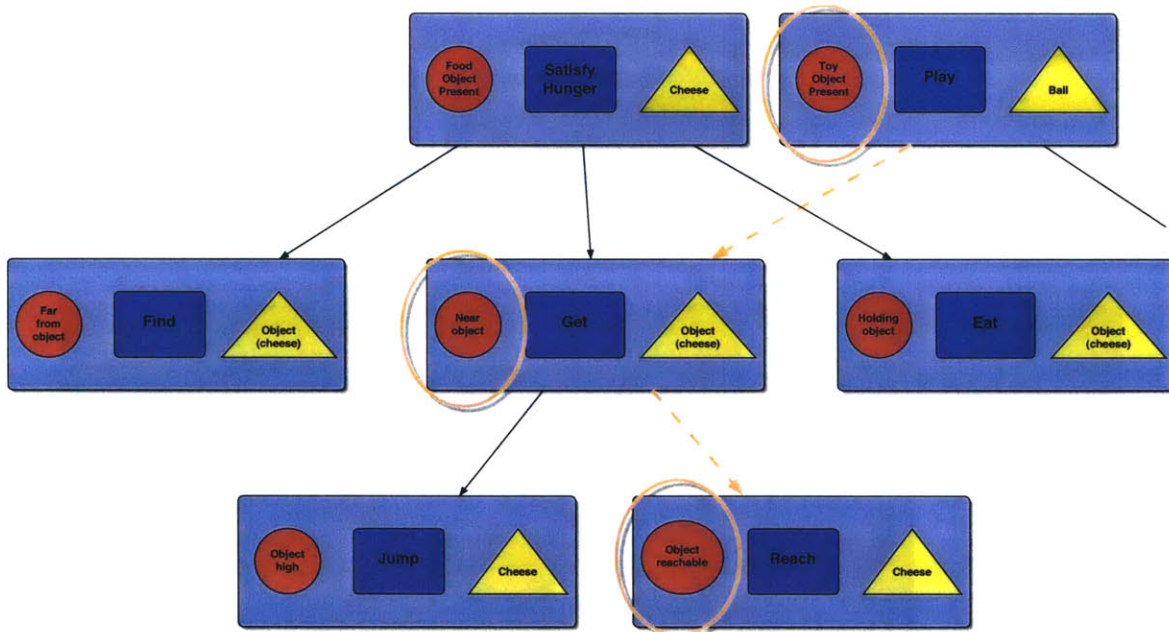


Figure 6-22: Max evaluates the *can-I* triggers along the path from the *play* tuple to the *reach* tuple

jump in order to get it, the trigger will return false. This means that Morris is not reaching in order to play, and so Max goes back to evaluate the other option, which was reaching in order to eat (shown in figure 6-23).

Here, all the *can-I* triggers come out true—there is a food object (cheese) present, it was ‘gettable’, and it was within reach. Max concludes that even though he reaches for toys more frequently than for food, under the circumstances, Morris was most likely reaching for the cheese.

If the ball and the cheese had been close together, where they were both reachable, Max would have mistakenly guessed that Morris was reaching for the ball rather than for the cheese. However, this is a ‘natural’ mistake—one person observing another reach towards a number of objects would have difficulty deciding which one was the desired object without additional information.

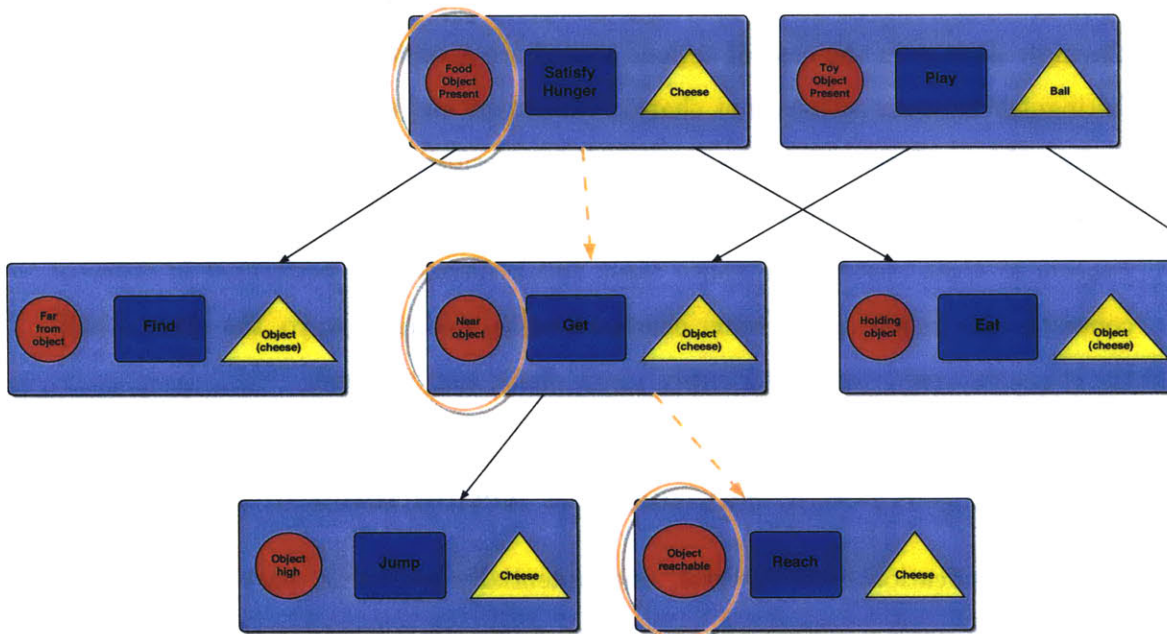


Figure 6-23: Max evaluates the *can-I* triggers along the path from the *satisfy hunger* tuple to the *reach* tuple

Distinguishing Actions that Share Movements

In many cases, different actions (i.e. actions that have different trigger or do-until contexts, or operate on different objects) utilize the same movement. In the previous example, *reaching for the cheese* and *reaching for the ball* could be considered different actions, because they are directed at different objects. Max can't tell which of these Morris is performing just by identifying the *reach* movement—he needs to evaluate the tuple and find Morris's *object of attention* to discover whether he's reaching for the cheese or the ball. Our flexible *belief selector* mechanism (described in section 5.5.1 allows the same action tuple to operate on multiple objects, so that the *reach for cheese* and *reach for ball* actions can be represented by the same tuple. However, if desired, they could also be represented as different tuples.

For example, a character might use the same movement for yo-yoing and for dribbling a ball. In this case, the character might have a *yo-yo the yo-yo* action and a *dribble the ball* action, which happen to perform the same movement on different objects. This might be particularly desirable if the two actions also have different

do-until contexts (e.g. yo-yo until you can ‘sleep’ the yo-yo vs dribble until done playing). Just as in the case of actions that share a tuple, the system can correctly distinguish between actions that share a movement and are represented in separate tuples. If one of our characters saw the other perform a yo-yo/dribble movement, he would identify all of his tuples that use that movement, and evaluate the *can-I triggers* in their hierarchies. Since these two behaviors are performed in different contexts, (one requiring the other character to be holding a yo-yo, the other a ball) the character could correctly identify which one it had observed.

6.2.3 Step-By-Step Summary of Action Identification

Action identification is an integral, and potentially confusing, part of this thesis. Here, I will provide a step-by-step summary of how Max is able to identify Morris’s actions.

Movement recognition. The very first thing Max must do is identify the *movement* he saw Morris performing, classifying it as one of his own movements.

Finding actions that use the matching movement. Once Max has identified the movement he saw Morris perform, he checks his *movement-action map* for all the leaf nodes in his action system that use that movement.

Paths through the action hierarchy. Next, Max finds all the paths through his action hierarchy which lead to these leaf action tuples. In other words, he identifies all the higher-level actions that invoke the potentially matching lower-level ones. He also identifies all the motivations that might lead to these actions, since they’re at the top of any path through the action hierarchy.

Evaluating the paths. Max needs to determine which of these actions (if any) it would be possible for Morris to be performing. He does this by evaluating the *can-I triggers* of the tuples all the way down the paths to the matching leaf actions.

Taking Morris’s perspective In order to correctly evaluate the *can-I triggers*, Max must evaluate them from Morris’s perspective. Normally, these triggers operate on Max’s belief about himself—where his body parts are, where objects are relative to him—here, he must use his belief about Morris instead.

Identifying the correct path. Max eliminates any paths that aren’t possible, and any leaf action-tuples that have no possible paths leading to them. When he is left with one path to one possible lowest-level action, Max concludes that this is the behavior Morris was performing.

6.2.4 Representing Self and Other

As alluded to in the previous sections, a critical part of identifying another’s actions is being able to use their point of view instead of one’s own—and being able to easily distinguish between the two. One of the most important beliefs in Max’s belief system is his belief about himself. Max’s self-belief contains all his proprioceptive knowledge (i.e. where he and his body parts are in space). All of the *triggers* and *do-untils* in Max’s system which rely on self-knowledge (e.g. triggers that rely on proximity to an object, do-untils that succeed when holding an object) operate on Max’s self-belief.

Similarly, Max has a belief about Morris which contains information about where Morris currently is, where his body parts are, which way he is facing etc. Importantly, Max filters his sensory information about himself and his sensory information about Morris through the same percepts. This means that the information in the two beliefs is stored in the same kinds of *percept histories*, and in the same format. Having the two beliefs in the same format allows Max to easily evaluate all the triggers and do-untils in his system that rely on self-knowledge from Morris’s perspective, simply by having them operate on his belief about Morris instead.

Max’s action system has a belief selector for his self-belief and another for his *reference character belief*—the character he is acting in reference to, or interacting with (like the *object of attention* this belief is optional, so that not all actions need to be in reference to another character). The *reference character* belief is used whenever

Max needs to interact with or attend to Morris. Max can take Morris’s perspective by swapping the beliefs returned by the selectors, before evaluating actions. This allows Max to evaluate actions “as if” he were Morris, while the independent beliefs and belief selectors guarantee that his knowledge of Morris and of himself remain separate.

6.2.5 Motivations and Goals

In the previous sections we focused on *can-I* triggers for action tuples—triggers that represent whether a particular action is possible under the current circumstances. Another subset of trigger contexts, known as *should-I* triggers, can be viewed as simple motivations—for example, a *should-I* trigger for Max’s eating action is hunger. Similarly, some do-until contexts, known as *success* contexts, can represent low-level goals—Max’s success context for reaching for an object is holding the object in his hands. By searching his own action system for the action that Morris is most likely to be performing, Max can identify likely *should-I* triggers and *success* do-untils for Morris’s current actions. For example, if Max sees Morris eat, he can match this with his own eating action, which is triggered by hunger, and know that Morris is probably hungry. Similarly, Max can see Morris reaching for, or jumping to get, an object, and know that Morris’s goal is to hold the object in his hands, since that is the success context for Max’s own *get* action. Notice that in this second case, Max does not need to discern the purpose of jumping and reaching separately, since these are both subactions of *get* in his own hierarchy.

We are currently developing mechanisms that allow Max to use the trigger and do-until information from his best matching action in order to interact with Morris in a more socially intelligent way—for instance, Max might see Morris reaching and help him get the object he is reaching for, bringing him closer to more advanced social behavior such working on cooperative tasks (this future work is discussed further in section 7.2.4).

6.2.6 Learning About Objects

One important way in which Max can already learn by observing Morris is through a process similar to that of *social referencing*, described in section 2.1. By watching Morris interact with unknown objects, Max can learn some of the affordances of these objects. Let's say Max starts out knowing that cheese is edible, but not knowing anything about ice cream. Meanwhile, Morris knows that ice cream is an edible (and tasty) treat. If Max watches Morris reach for the ice cream and is asked to identify what Morris is doing he will shrug, indicating that he doesn't know why Morris is reaching. This is because none of the possible paths to the *reach* tuple in Max's action system seem valid (see figure 6-20)—there are no toy or food objects that Max knows about within Morris's reach, and Max doesn't know what the object within reach is for.

If however, Max sees Morris eat the ice cream cone, the story is different. When Max sees Morris eat the ice cream cone, he tries to identify the action as usual—first he identifies the movement he saw Morris perform as *eating*, next he identifies the bottom-level action tuples *eating* is a part of, finds the *eat* tuple, and traces back up the hierarchy from the *eat* tuple to the *satisfy hunger* tuple. Finally, he evaluates the *can-I* triggers along the path between *satisfy hunger* and *eat*, and finds that they are not satisfied—there is no food object to eat.

At this point, Max notices something important—the *eat* action tuple (and in turn the *eating* movement) is only ever used to satisfy one drive, because it is only part of one motivational subsystem (Max can determine this by looking at the number of top-level tuples he has traced back to). Since eating to satisfy hunger is the only purpose Max knows of for the *eating* movement, he checks to see if Morris could have been eating an unknown object. To do this, he re-evaluates his *can-I* triggers with a slight modification. He replaces the first *can-I* trigger—*food object present* with another trigger, one that selects the object Morris is most likely to be interacting with. The simplest version of such a trigger just picks the object that was closest to Morris, though versions that take gaze direction into account have also been used.

Once this object has been selected, Max checks to see whether the remaining *can-I* triggers have become true. In this example, Max would choose the ice cream as Morris's likely *object of attention*, and would find that Morris was in fact holding the ice cream, making it possible for him to be eating it. Max would conclude that Morris was eating the ice cream, and would add the *ice cream shape* to his *food percept's* list of edible shapes. From this point on, Max would recognize ice cream as a potential food source.

Max can learn about unknown toys in a similar manner. Here, there is a bit of a twist, because toys can be played with in two different ways—by dancing or throwing (the *play* motivational subsystem of the action hierarchy is shown in figure 6-24). This means that when Max sees Morris dancing with a new toy he must learn both

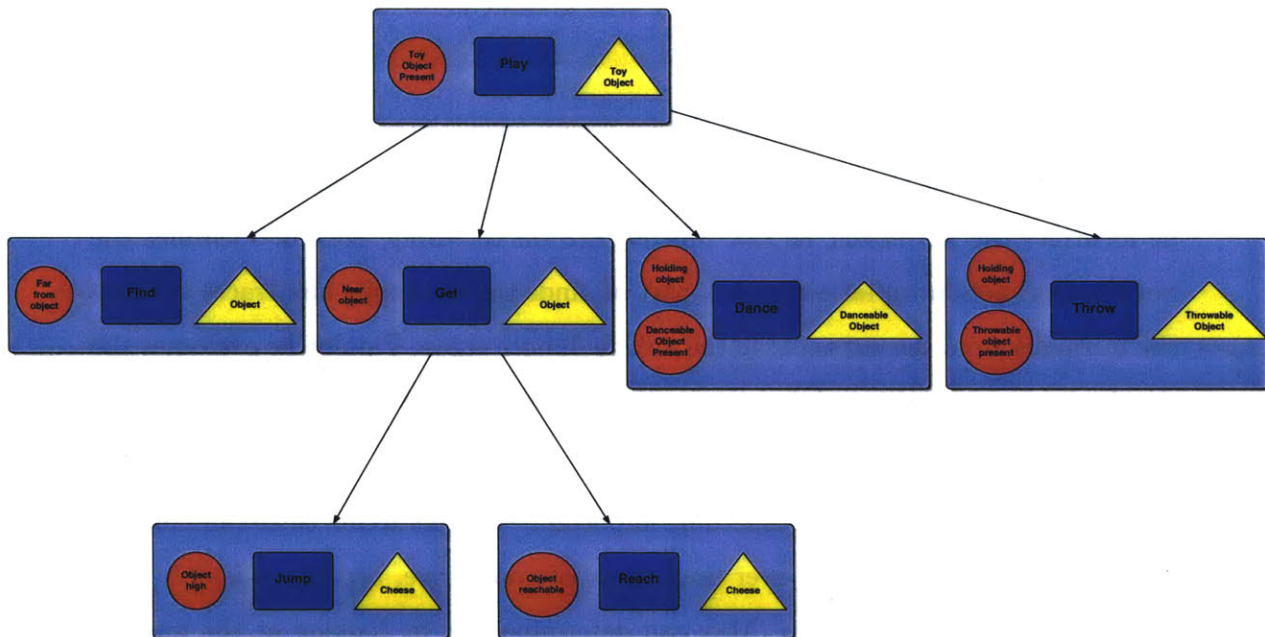


Figure 6-24: The *play* motivational subsystem of the action hierarchy. Notice that the *dance* and *throw* tuples have more than one *can-I* trigger—a *proximity trigger* (*holding object*) which checks whether the *object of attention* is in Max's hands, and an *object selection trigger* (*Danceable object* and *Throwable object*), which checks whether the *object of attention* is of the appropriate type for that tuple to act on.

that it is a toy and that it is 'danceable'. Max does this by checking all the *object selection* triggers in the matching path through the action hierarchy. Each *object*

selection trigger checks for objects marked by a particular percept, and Max adds the unfamiliar object's shape to each of these percepts. So, in the case of seeing Morris dance with a hoop, Max would add the *hoop shape* to both the *toy object* and *dance object* percepts, which are connected to the *toy object present* and *danceable object present* triggers respectively.

6.3 Results

In this section, I would like to provide a brief summary of Max's social learning capabilities, and present some additional figures demonstrating these abilities.

What Max Can Do

- Max is able to observe Morris using synthetic vision, and parse the continuous stream of motion he observes Morris performing into individual movements.
- Max can use his own movement representation to identify the observed movement. In other words, he classifies the observed movement as one of his own—a form of perception-production coupling.
- Similarly, Max can use his own action system to identify the action he thinks Morris has just performed. This identification occurs across the full range of granularity in Max's action hierarchy.
- Identifying the action he believes Morris performed allows Max to also identify the motivations and goals of that action, represented as *should-I triggers* and *success do-untils*.
- Max can learn about the affordances of objects such as food and toys by watching Morris interact with these objects.

What Max Can't Currently Do

- Max does not learn the correspondence between his body and Morris's—this knowledge is built into the system.
- As mentioned in the introduction, Max cannot currently identify the movements of characters with a different morphology than himself, and thus cannot identify their actions and motivations.
- While Max is capable of using an approach similar to the one described in this thesis to identify emotions, this extension is not yet fully implemented.
- Currently, while Max can identify Morris's goals, he does not act on this knowledge.
- At the moment, Max simply ignores movements and actions he does not recognize. Once again, functionality for learning these actions rather than ignoring them already exists in the system, but it is not currently being utilized.

The following chapter presents future work meant to address a number of these issues, as well as further discussing the results and implications of this thesis.

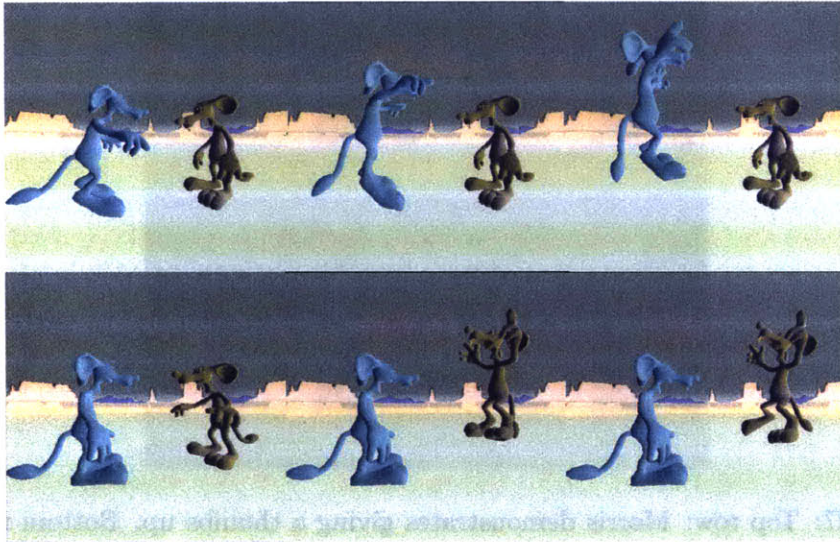


Figure 6-25: Top row: Morris demonstrates jumping. Bottom row: Max imitates jumping.

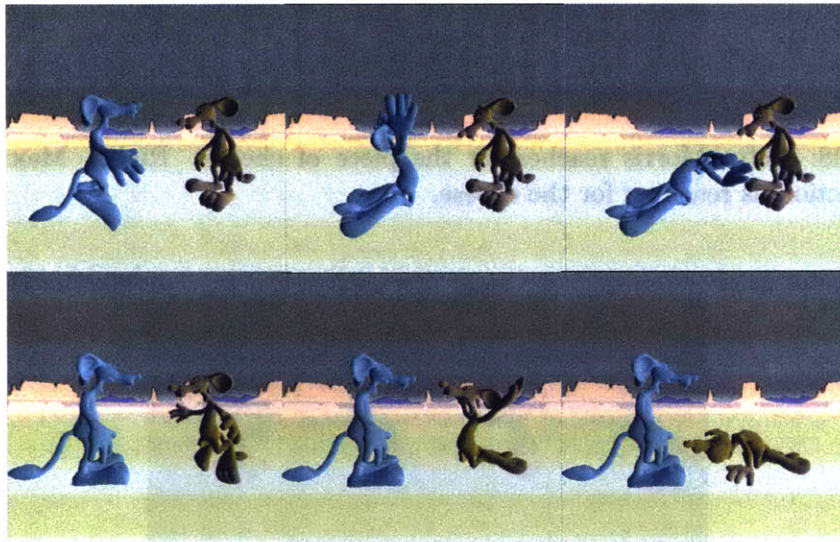


Figure 6-26: Top row: Morris demonstrates pounding the ground. Bottom row: Max imitates pounding the ground.

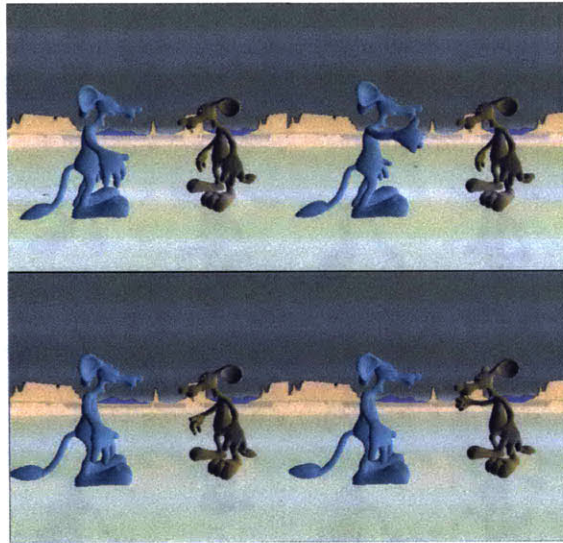


Figure 6-27: Top row: Morris demonstrates giving a thumbs up. Bottom row: Max imitates giving a thumbs up.



Figure 6-28: Left: Morris reaches for the piece of cheese. Right: Max identifies Morris's action as reaching for the cheese.



Figure 6-29: Left: Morris eats the piece of cheese. Right: Max identifies Morris's action as eating.

Chapter 7

Discussion and Future Work

At this point, we have seen that Max the Mouse (and by extension, other animated characters and robots developed using the Synthetic Characters architecture), is capable of imitating other characters, identifying simple motivations and goals for their behavior, and learning from their actions. Furthermore, the implementation of these abilities relies strongly on mechanisms and approaches suggested by the cognitive literature, such as motivationally-driven, hierarchical action structures, perception-production coupling, and Simulation Theory. In the following sections, I will evaluate and discuss these results, exploring their implications for cognitive research, and for future work in socially intelligent artificial creatures.

7.1 Stumbling Blocks, Successes and Surprises

In the course of developing the characters and interaction scenarios presented in this thesis, a number of interesting phenomena have come to light. First, and most importantly, a cognitively-inspired, and in particular, simulation-theory inspired, approach to social learning in synthetic characters has proven to be extremely effective, and there is high hope for its extensibility to characters with further social learning skills and greater social intelligence. Second, using a hierarchical action system contributes much of the ease **and** much of the challenge in implementing imitative behavior. Third, many of the seemingly separate social learning phenomena described in the

cognitive literature may be relatively easily achieved once a number of core mechanisms (e.g. hierarchical action system, movement recognition, Simulation Theory) are in place. Finally, the problem of movement recognition is a significant challenge in interpreting observable behavior, but can be noticeably simplified by using the right motor representation and body-knowledge. The next few sections will address each of these issues.

7.1.1 Simulation Theory as the Road to Social Characters

Simulation Theory is in many ways the unifying factor among the various social learning tasks and mechanisms tackled in this thesis. I chose to consistently use a simulation-theoretic approach while addressing a wide variety of social learning problems, in part because of the strong supporting evidence from cognitive and neuroscientific research (discussed in section 2.2.1), but also in order to see just how far an artificial creature could get using itself as a model for other's behavior. As it turns out, the answer is pretty far, and certainly this thesis has not hit the limit. From recognizing observed movements, to finding another character's object of attention, to identifying another's motivations and goals, Simulation Theory has proven to be an effective approach to a range of social learning problems, and perhaps more importantly, as discussed in the following sections, it is often an approach that simplifies the problem at hand.

Movement Recognition and Motor Representation

Motion parsing and movement recognition from visual data are extremely challenging problems, and currently represent very active research areas. In particular, on-line movement classification systems typically require a large set of training data, and rely on statistical models to extract motion features from the data that correlate with particular gestures (but see [16]). However, by using their own movement repertoires as the example set, our characters are able to perform on-line movement classification without any training period, and using only a limited set of body part coordinates

(rather than joint angles or statistical models) as input, which to my knowledge has not been done before (the current limitations of this approach, as well as possible extensions, are discussed in sections 7.2.1 and 7.2.2).

Advantages of Self-Knowledge: Using Perception-Production Coupling

In the case of movement recognition, and also in general, Simulation Theory allows for an elegant conservation of representation. The body knowledge that the character needs in order to identify a movement (where different body parts go, and how they move) is knowledge **it already possesses** in order to execute that movement. At the heart of perception-production coupling is the powerful idea that knowledge necessary to executing your own actions can be reapplied to interpreting the actions of others, removing the need for separate ‘other modeling’ machinery.

For instance, other systems that parse motion capture data into higher level action sequences, such as Bindiganavale’s work [16], often require that spatial and movement constraints (e.g. ‘my feet can’t go through the floor when I walk’) be explicitly represented in the constructed action before it is reproduced by another character. When one of our characters imitates another walking, he knows that his feet can’t go through the floor, because that is already a necessary piece of information for executing his own walking behavior. In general, the character will automatically apply any constraints on his own behavior to interpreting behaviors it observes. (However, unlike Bindiganavale’s system, this system does not currently address characters of different morphology—this is discussed in section 7.2.1).

The Role of Motivationally-Driven Hierarchical Action Structures

in more cognitively complex tasks, the advantage provided by perception-production coupling can be even more striking, particularly when applied within a hierarchical action structure. For instance, when a character such as Max maps one of his action tuples onto the observed behavior of another, he is not only provided with the information contained within that tuple—the action to be performed, the immediate goals of that action, and the environmental context that triggered it—but with all

the information contained within the hierarchy that tuple belongs to.

When Max see Morris reaching for the cheese, and identifies this with his own reaching tuple, he knows not only that Morris is reaching for the cheese, but that he wants to get the cheese—if the cheese were higher up Morris might jump instead of reach, but both have ‘getting’ as their goal. Looking still higher up the hierarchy tells Max that Morris wants to get the cheese because he is hungry, and that he might want other food items as well.

Motivationally-driven hierarchical action structures provide an ideal cognitive substrate for Simulation Theory, because they so neatly package together the key characteristics of an action—the movements involved, the motivations for the action, and the actions goals (a slight variation on Call and Carpenter’s three sources of information, described in section 2.1). Further, hierarchical structures facilitate not only the recognition of immediate goals and motivations, but also recognition of the hierarchy of goal-directed behavior that characterizes intentional action (see section 2.2.5). In other words, using a hierarchical action structure produces a hierarchical intention structure, and allows such a structure to be recognized in others (see section 2.2.5).

Building Blocks of Social Learning

At the beginning of chapter 2, I introduced the different categories of social learning described in the cognitive and ethological literature. I also discussed a number of theories that suggested that these apparently different types of social learning might result from responding to different aspects of the stimuli or represent different uses of the same underlying mechanisms and structures.

The work in this thesis seems to strengthen the case for shared social learning mechanisms. At least in the artificial system presented here, it appears that, given a number of key mechanisms—namely movement recognition, a motivationally-driven hierarchical action structure and the ability to simulate another’s point-of-view on that structure, a large number of seemingly disparate social learning abilities can be demonstrated. To give a few traditional examples:

Stimulus Enhancement While this was not a skill we focused on particularly, Stim-

ulus Enhancement is easily accomplished in this architecture. In the system presented here, the other character's *object-of-attention* is always identified. By subsequently adopting this object as its own object of attention, our character would demonstrate stimulus enhancement.

Mimicry and Movement or Action Level Imitation Many different terms have been used for the simple reproduction of the movements produced by another. Whatever you want to call it, it has been achieved (and described in detail) in this thesis.

'True' Imitation Whenever Max demonstrates what he believes Morris to be doing, he copies not only the form of the movements, but the object they are aimed at, and what he believes to be their goal.

Emulation Again, while it was not explicitly addressed in this thesis, the characters presented here are capable of emulating the results (or goals) of an action. They would do this by focusing only on the matching *do-until context* of that action, rather than on the action as a whole.

Identifying Goals and Motivations Perhaps most importantly, hierarchical action systems help a character to identify another's motivations and goals at multiple levels of granularity.

The ability of this system to potentially reproduce all of these forms of social learning seems to change the critical question from "which kind of social learning is occurring?" to "which kind of social learning is most appropriate here?". This is not a trivial question. What the previous list shows us is that attending to different aspects of the identified action, different levels of the hierarchy, leads to different responses. How to tell whether the movement, or the object, or the result of the action is the critical part, whether the immediate goal or its parent is most relevant, are important and unresolved questions.

Unfortunately, success in an artificial system cannot definitively prove anything about natural systems—it can only suggest. Nevertheless, the success of this ap-

proach, coupled with previous ethological research pointing to the existence of hierarchical action systems in animals, lends support to the idea that differences in social learning abilities may represent differences in which levels of the hierarchy different animals imitate (in the broadest sense of the word), and which sources of information they attend to.

7.1.2 Limits of Simulation Theory

The Importance of Representational Choice

While there are many obvious benefits provided by perception-production coupling, to some extent, it is also a double-edged sword, since it makes the choice of how to represent the character's self-knowledge doubly important. For example, how movement is represented and produced by the character's motor system can profoundly influence how easy it is to use that same representation to recognize another's movements.

Let's say that instead of a motor system made up of example animations (or movement primitives) we had a motor system that generated animations procedurally, through a combination of inverse kinematics and physics simulation. It's not entirely clear how such a representation could even be used to recognize purely visual input. At the very least, we'd need knowledge of the other character's joint structure, and of the physical forces impinging upon them.

Substituting Knowledge of Another for Knowledge of Oneself

Similarly, in this thesis, it was fairly challenging to design action tuples in such a way that they could easily be evaluated from multiple perspectives. There are many subtleties to taking someone else's point of view—it is not enough to simply pretend for a moment that the other creature is in your body and leave it at that. In order for one creature to really simulate another, they must imagine themselves in the other creature's location, looking where they are looking, and perceiving what they perceive. More than that, they must remember to think about objects in the world as they relate to the other **not** as they relate to themselves. Finally, even their memory

must be from the other creature’s perspective and not their own—what the other was doing previously, the object they were near 2 seconds ago.

In other words, it is very easy for pre- and post-conditions, motivations and goals, to implicitly assume the point-of-view of the simulator, and therefore for a Simulation Theoretic approach to fail. Just because using the same representation for perception and production can be easier doesn’t always mean it has to be. **Critical to the success of a simulation-theoretic approach is the ease with which a character’s knowledge of another can be substituted for his knowledge of himself.**

What Simulation Theory Can’t Do, and the Need for a Complete Cognitive Architecture

Simulation Theory’s greatest benefit—economy of representation—is also its greatest weakness. A purely simulation-theoretic approach would mean that a character can only understand what it already knows how to do itself. For instance, our character’s motor systems now represent not only the space of movements they are capable of producing, but the space of movements they are capable of **understanding**. Similarly, the actions our characters can recognize are limited to the ones they already know how to perform, and the goals they can recognize are limited to the ones they can try and achieve (but see sections 7.2.6, 7.2.7 and 7.2.8 for ideas on how our simulation-theoretic approach can be used to learn new movements and bootstrap the understanding of new actions and goals). In part, this is a limit of our current system - if our characters had mechanisms for generating new goals perhaps they could use these structures to generate potential goals for the other character.

However, this solution makes an important point—in many ways, Simulation Theory takes the hard problem of theory of mind, and pushes it out into the rest of the cognitive architecture. **The character’s ability to understand complex behavior rests on its ability to generate complex behavior.** For humans and other animals this is in general a good deal, since they must be able to generate these same behaviors in order to survive. Similarly, since the Synthetic Characters system contains a complete cognitive architecture, it was relatively easy to take advantage of the

already existing cognitive structures to create social learning behaviors. However, it brings up the question of whether robotic and animated systems designed expressly for the purpose of interpreting and reacting to human behavior, and thus lacking a general cognitive framework, could be as successful.

Perhaps the most profound limitation of a Simulation Theoretic approach (at least in a motivationally driven system) is the inability to understand unknown motivations. As discussed in section 2.2.1, Simulation Theory rests on the assumption that the other is “like me”, and therefore does things for the same reasons I do. While novel goals frequently generate novel results, and novel movements and actions are visually different from known ones, novel motivations are potentially invisible. Thus, even if a creature has cognitive mechanisms for generating new motivations within itself (for instance, I currently have a motivational drive to finish this thesis, which presumably I was not born with), it is not clear that these could be used to understand an unknown motivation behind another’s actions, since it would be hard to even recognize the motivation as novel, or determine which actions were being used to satisfy it (but see my discussion in appendix B regarding the fuzziness of the distinction between motivations and goals).

Of course, it isn’t clear that even humans (let alone animated characters) are very good at understanding others who are radically different from themselves. To the extent that we are capable of understanding motivations that differ significantly from our own, other powerful mechanisms, such as language, may come in to play.

7.2 Future Work

While the characters described in this thesis are capable of a number of complex social learning tasks, in many ways they have just begun to scratch the surface of socially intelligent behavior. In the following sections, I discuss the limitations of this work, and illuminate some of the ways in which the work presented in this thesis could be extended and used to bootstrap additional social learning skills.

7.2.1 Solving the Correspondence Problem

Currently, the characters in this thesis have a completely hard-wired knowledge of the correspondence between their own body parts and those of others. Additionally, the two characters presented in this work have identical morphology, an assumption that makes movement recognition easier, since it allows observed movements to be fairly directly compared to those in the character's motor system, without having to compensate for different proportions. Perhaps more importantly, the characters in this thesis have easily observable and identifiable body parts (e.g. hands, feet, knees etc. that can be tracked).

The problem of learning the correspondence between bodies with differing morphologies is an important one to address if we want to develop characters (and robots) capable of observing, imitating and identifying human action. In order to imitate a human, characters (and robots) must be able to map the actions they see the person performing onto their own actions, even though the person is shaped differently from themselves. I'll address the problems this presents our system in order of complexity.

Ideally, in order to recognize human movements, the character would be able to observe the person visually, and extract key body part locations from the image (this is of course currently an ambitious visual processing goal for real-time interaction). If the character and the model are similarly proportioned (e.g. the demonstrator is much smaller or larger, but has proportionally similar arms and legs), then the root-relative positions of their body parts could be scaled and mapped onto the character's morphology, to create movements that are comparable to those in the character's repertoire.

If the demonstrator has different proportions from the character we will need to use an approach that incorporates new information, present, but not used, in this thesis, in order to learn the mapping from their body to the character's. In this case, the character could potentially map actions it observes onto its own motion space by looking not only at the root-relative positions of body parts, but at their positions relative to each other (and perhaps to objects in the environment). For instance, if

Morris had a much longer neck than Max, Max could note that when Morris covers his eyes his hands are next to each other on his face, rather than simply noting how far his hands have moved from the center of his body (which on short-necked Max, might translate to way over his head). Similarly, the distance metric for comparing observed movements to movements in the character’s system would have to take distances between body parts, and not just between body parts and the root node, into account.

Another approach, one that can be used with visual data or data from motion capture suits (which is, at present, a more realistic source for human data), is to learn a model of the correspondence between the demonstrator’s and character’s bodies using an interactive training period (this is the approach we used in [28], and used by Lieberman in [75]). Here, the character performs a series of movements and assumes that the person (or other character) is imitating them. They can then use one of a number of standard machine learning techniques (for example neural networks [28] or RBFs [75]) to develop a model of the correspondence between the observed input (e.g. body part positions or joint angles) and the movement they just performed. This approach has the advantage of working with many different forms of data input, and of mimicking the turn-taking imitation games engaged in by infants and caregivers. It also allows the character to learn correspondences to demonstrators with very dissimilar morphologies (e.g. dolphin vs human), though it relies on the demonstrator to come up with a good imitation of the character’s action. Importantly, once a model is developed, it can be used to represent subsequent observed movements by mapping them onto the creature’s own body. This means that the search for the best-matching movement, including the distance metric, would not need to be altered.

7.2.2 Other Problems in Imitating Humans

Ultimately, we would like to create synthetic characters who are able to imitate and understand human action. However, there are a number of additional challenges that using human motion capture data would present. In general, motion capture data, particularly data gathered through visual analysis (as opposed to via a motion capture

suit) is significantly noisier than data gathered through synthetic vision. It remains to be seen whether the movement parsing and recognition techniques presented here would hold up to greater signal noise—though the system’s ability to compensate for obstructed data points leads us to be optimistic on this front.

Another problem presented by motion capture data is the possibility of differing frame rates between the character’s motion and the observed human motion. This can be compensated for using techniques such as interpolation and dynamic time warping, but again, we will need to see how accurate our movement recognition continues to be under these conditions.

Motion Parsing

The “hub-and-spoke” model of movement parsing presented in this thesis is a relatively elegant solution to breaking a stream of animated motion into individual movements. Since a similar approach has been used on human motion capture data [56], we hope that this method can be generalized, in order to allow our characters to parse data from human motion as well. The primary challenge there would be in identifying and recognizing the hubs in the first place, which remains an uncharted research area (previous work has used user-identified hubs).

In the case of human motion capture data, the supplementary motion parsing techniques mentioned in section 6.1.2 may become more important. For example, using similar approaches (e.g. looking at end-effector position, and changes in movement quality) Lieberman [75] has successfully implemented a system for parsing human movement from a motion capture suit.

7.2.3 Understanding Emotions

Another way in which the system presented in this thesis is currently being extended, is through the creation of characters who can recognize and respond to each other’s emotions. This thesis has primarily focused on goal-directed actions—actions which try to to satisfy a particular motivation or carry out an intention, generally towards

an object in the world. However, humans often perform actions that are instead emotionally communicative—conveying a particular affective state. People smile, shrug, give a thumbs up, cross their arms and frown, wring their hands and so on. Correctly interpreting these sorts of affective gestures is a critical part of human social interaction [69]—emotion recognition is even considered a significant predictor of social competence in children [74].

The characters Max and Morris already have a large repertoire of emotionally significant actions, and a number of autonomic variables devoted to their current emotional state. By applying the action identification techniques described in this thesis to emotionally driven portions of their action systems, Max and Morris could use simulation theory to identify each other’s emotions.

For instance, if Max brought Morris a piece of cheese, Morris could respond with a positive gesture, such as giving a thumbs up, or a negative gesture, such as covering his face in frustration, or crossing his arms and tapping his foot. Max could search his own action tree for the emotions that would cause him to display these behaviors, and know whether Morris was pleased or displeased with his offer. Further, by using Simulation Theory, he could quickly identify the affective content of many different gestures, rather than having to learn for example, that both a ‘thumbs up’ and a ‘joy dance’ are positive gestures. A Simulation Theory-style understanding of emotional displays would be an important part of developing cooperative behavior between characters. For related work on agents who interpret and display emotion see for example Breazeal’s work with Kismet [26], Picard’s Affective Computing research [92], and Cassell’s Gesture and Narrative Language research [37].

7.2.4 Cooperative Behavior

One critical aspect of human social behavior is our ability to cooperate and work on joint tasks. The American Heritage Dictionary defines cooperating as “to work together towards a common end or purpose” [2]. Thus, in order for one character to engage in cooperative behavior with another it must be able to recognize and adopt the other’s goal.

The first step of cooperative behavior, recognizing the other’s goal, has already been accomplished (at least at a low level) in this thesis. Max is able to identify the goals of Morris’s actions by looking at the goals of his own actions. We are currently implementing the second step—acting on this goal in order to help accomplish it.

Here we are faced with a potentially tricky situation, which I will use an example to illustrate. Let’s say that Morris’s goal (that is the *success do-until* for his action) is to get himself the cheese. In order to help Morris, Max must now act in a way that is both different from how he would get the cheese for himself, and how he would get the cheese if he were Morris. In both those cases, Max would simply go get the cheese for himself, which would not help Morris at all.

Simulation Theory has helped Max identify Morris’s goal which is “get myself the cheese”, but in order to act correctly Max must now alter that goal slightly before acting on it himself—it must become “get **Morris** the cheese”. Now, Max will be ready to identify (or construct) an action that satisfies this goal.

Finally, there are multiple levels of goals in the action hierarchy, and it will be an interesting question to explore which ones to help satisfy when. For instance, instead of trying to get Morris the out-of-reach cheese, Max could recognize Morris’s hunger, and get him some ice cream instead.

7.2.5 Predicting Future Actions

Another piece of functionality that already exists in the system, but has not yet been put to use, is the ability to predict other character’s future actions. The simplest way to do this would be for a character to evaluate their action hierarchy using conditions that assume the success of the identified action. For example, when Max see Morris reaching for the cheese, he knows that the success context of reaching for the cheese is holding the cheese. Max could now check which of his own action tuples has a *can-I trigger* satisfied by holding the cheese, and would discover that the *eating action* does. He could then predict that, once Morris gets the cheese, he’s likely to eat it. This approach could potentially be made even more accurate by having the character keep track of which actions tend to follow which (an action-action map similar to

the movement-action map already in use), and using this information to help with ambiguous predictions.

7.2.6 Learning New Movements

For simplicity’s sake, in this thesis, Max simply ignored any observed movements he didn’t recognize. However, characters using the architecture presented in this thesis have been developed who create new *movements* from unrecognized observed gestures, and add them to their movement repertoire. One ongoing issue in this area is deciding whether a movement is new or simply a relatively poor match to an existing movement. We have previously addressed the problem of modeling novel movements and distinguishing them from known movements, and plan to apply the approach described in Blumberg 2002 [21] here.

7.2.7 Learning New Actions

The trickiest part of learning new movements isn’t learning how to perform them, it’s learning when to use them. As discussed earlier (see section 3.2.1), imitation has often been seen as a potential way to quickly teach characters or robots what movements (or combinations of movements) to perform in which situations. Currently, Max can learn new objects to apply existing actions to, by watching Morris interact with these objects. We would like him to be able to learn new actions to perform by observing Morris as well.

If a movement, or series of movements is executed in response to an unfamiliar environmental context, and generates an unfamiliar result, it may represent a new action. If the result is a desirable one—perhaps one that already matches the goal state of one of the observing character’s existing actions—it may be worth while to construct a new action tuple based on this observation. So for instance, let’s say that Max saw Morris use a tool, such as a rake, to bring food closer to himself. Using the raking movement in the context of out-of-reach food may not be familiar to Max, but the result—holding and eating the food—is. He could then construct an action tuple

that requests a raking movement, and is triggered by the presence of a rake and of out-of-reach food, and has holding the food as its do-until context.

There are of course some subtleties of implementation here, that need to be explored. Identifying the results and triggering contexts of an action, as distinct from other aspects of the changing environment, is a challenge. Simulation Theory may potentially help here as well, allowing the character to focus on the kinds of environmental events and changes it normally finds most salient.

7.2.8 Learning New Goals

Perhaps the most critical step in intelligent social behavior (and in intelligent behavior in general) is the ability to adopt new goals and sub-goals. Our characters could potentially learn new goals to pursue by watching others act. If another creature repeatedly uses actions to achieve the same novel result, this result may represent a new goal.

As mentioned in the previous section, it can be difficult to pick out the result of an action from other changes in the environment. This problem is compounded in the case of new goals, since actions are not always successful, and the result of an action may not necessarily be the desired one.

Our characters can mitigate these problems by applying their other social skills to the task of understanding novel goals. In particular, this is a situation where affective feedback and social referencing may be particularly critical. A result is unlikely to be the desired one if the other character appears unhappy with it, but if the demonstrator is happy with the result, they were probably intending to achieve it. Similarly, when the imitating character tries to adopt the new goal it can look to the model for approval, to see if it has the right idea.

The Future—Towards Characters who want to Learn, and Demonstrators who want to Teach

In general, one of the fundamental features of human social interaction and social learning is that it is almost never one way—instead, human social interaction is characterized by turn-taking, feedback and reciprocity. A character who must learn by observing an oblivious demonstrator is relatively limited in what it can discover. On the other hand, characters who are aided by the presence of a knowledgeable demonstrator will have many additional opportunities to learn available to them (see Breazeal *et. al.*'s work on interactive tutelage for humanoid robots [29]). Our ultimate goal for the future then, may be to develop synthetic characters and robots who are capable of taking full advantage of what others (particularly humans) want to teach them. A Simulation Theory-based social learning system, that allows such characters to correctly interpret and respond to the behaviors of those they interact with, will be an important first step towards this type of socially intelligent, and socially responsive, artificial creature.

Appendix A

Synthetic Vision

Note: This appendix is adapted from Isla 2001 [65].

Synthetic Vision renders the world from the character’s point of view. The location of a visible object can be extracted visually by examining the screen-space coordinates of the centroid of the object in the point-of-view rendering. This 2-vector combined with depth information from the rendering’s depth buffer yield the NDC-coordinates of the object (see [48] for a discussion of NDC-space and camera projections). These coordinates can be converted into the local space of the camera (and of the observing creature’s eye) through the inverse-NDC transformation.

Assuming that the x and y NDC-coordinates range from -1 to 1, and the z NDC-coordinate ranges from 0 (at the eye-position) to 1 (infinitely far), and assuming that the camera projection properties are given by a frustum defined by f_{near} , f_{far} , f_{left} , f_{right} , f_{top} and f_{bottom} , the NDC-to-Local transformation is given by the following equations:

$$z_{local} = \frac{(-f_{near} * f_{far})}{(f_{far} - f_{near}) * z_{ndc} + f_{far}} \quad (\text{A.1})$$

$$x_{local} = \frac{z_{local}}{2f_{near}} * (f_{right} * x_{ndc} - f_{left} * x_{ndc} - f_{right} - f_{left}) \quad (\text{A.2})$$

$$y_{local} = \frac{z_{local}}{2f_{near}} * (f_{top} * y_{ndc} - f_{bottom} * y_{ndc} - f_{top} - f_{bottom}) \quad (\text{A.3})$$

Appendix B

Terminology

Throughout this work I have used a number of potentially ambiguous terms. While there are any number of official (and differing) definitions for these words, I would like to briefly describe how they have been used in this thesis. Words that are listed together have been used relatively interchangeably.

Movement, Motion, Gesture A movement, motion or gesture refers to a motor pattern played out on a creature's body—the series of muscle (or motor, or animated muscle) movements carried out by that creature, without reference to their context. Examples of movements from this thesis include: *jumping*, *reaching* and *covering the eyes*.

Action An action is a movement, or series of movements, placed in an environmental and motivational context. That is, an action occurs in response to certain (internal or external) circumstances, and generally has a desired result associated with it. Examples of actions include: *jumping for the cheese*, *reaching for the ball*, *covering eyes in frustration*.

Intention, Goal An intention or goal is a desired result the character plans to try and achieve, often associated with the actions the character plans to use to achieve it (which can then be described as *goal-directed*). Examples of goals are: *getting the baton*, *eating the cheese* and *getting close enough to the ball to reach it*

Motivation, Drive Motivations and drives are the ‘why’ behind intentions and goals. They are the reason for wanting the baton or eating the cheese. Examples of motivations and drives are: *Satisfying hunger*, *desire to play* and *desire to socialize*.

To clarify further, let’s take the simple example of Max reaching for the cheese. In this situation, Max’s **movement** is reaching, his **action** is reaching for the cheese, his **goal** is to get the cheese, and his **motivation** is that he’s hungry.

The distinction between motivations and goals to some extent melts away as we go farther up the action hierarchy. As actions become coarser in granularity, the difference between an intention and a drive appears to merge—at the top level one could argue that satisfying hunger is both a goal and a motivation. While this is an interesting point, teasing out the differences between goals and motivations is beyond the scope of this thesis, so I will simply note the potential vagueness of the current definitions, and leave it at that.

Bibliography

- [1] *Behavioral and Brain Sciences*, 21(5), 1998.
- [2] *The American Heritage Dictionary*. Houghton Mifflin Company, 4th edition, 2000.
- [3] O. Arikian and D. A. Forsyth. Interactive motion generation from examples. In *Proceedings of the 29th annual ACM conference on Computer Graphics and Interactive Techniques*. SIGGRAPH, 2002.
- [4] C. Atkeson and S. Schaal. Learning tasks from single demonstration. In *IEEE International Conference on Robotics and Automation*, pages 1706–1712. ICRA '97, IEEE, 1997.
- [5] C. Atkeson and S. Schaal. Robot learning from demonstration. In *International Conference on Machine Learning*, pages 12–20, 1997.
- [6] N. Badler. *Temporal scene analysis: Conceptual descriptions of object movements*. PhD thesis, University of Toronto, Department of Computer Science, 1975.
- [7] N. Badler, J. Allbeck, L. Zhao, and M. Byun. Representing and parameterizing agent behavior. In *Proceedings of the IEEE Conference on Computer Animation*, pages 133–143, Geneva, Switzerland, 2002. IEEE Computer Society.
- [8] N. Badler, M. Palmer, and R. Bindiganavale. Animation control for real-time virtual humans. *Communications of the ACM*, 42(8):64–73, August 1999.

- [9] J.A. Baird and D.A. Baldwin. Making sense of human behavior: action parsing and intentional inference. In B.F. Malle, L.J. Moses, and D.A. Baldwin, editors, *Intentions and Intentionality: Foundations of Social Cognition.*, pages 193–206. MIT Press, April 2001.
- [10] D.A. Baldwin and J.A. Baird. Discerning intentions in dynamic human action. *Trends in Cognitive Sciences*, 5(4), April 2001.
- [11] A. Billard. Imitation: A means to enhance learning of synthetic proto-language in an autonomous robot. In Dautenhahn and Nehaniv [42], pages 281–310.
- [12] A. Billard and K. Dautenhahn. Grounding communication in autonomous robots: An experimental study. *Robotics and Autonomous Systems*, 24(1-2):71–81, 1998.
- [13] A. Billard, K. Dautenhahn, and Hayes. G. Experiments on human-robot communication with robota, an imitative learning and communicating doll robot. In *Proceedings of Socially Situated Intelligence Workshop*, number CPM-98-38 in Center for Policy Modeling Technical Report Series. Fifth Conference on the Simulation of Adaptive Behavior, 1998.
- [14] A. Billard and S. Schaal. A connectionist model for on-line learning by imitaiton. In *Proceedings of the 20001 IEEE-RSJ International Congerence on INtelligent Robots and Systems*, Maui, HI, 2001. IEEE/RSJ.
- [15] R. Bindiganavale and N.I. Badler. Motion abstraction and mapping with spatial constraints. In *Modelling and Motion Capture Techniques for Virtual Environments, International Workshop Proceedings*, Geneva, November 1998. CAPTECH'98.
- [16] R.N. Bindiganavale. *Building Parameterized Action Representations from Observations*. PhD thesis, University of Pennsylvania, Computer and Information Science, 2000.

- [17] E. Bizzi, F.A. Mussa-Ivaldi, and S. Giszter. Computations underlying the execution of movement: a biological perspective. *Science*, 253:287–291, 1991.
- [18] B. Blumberg. *Old Tricks, New Dogs: Ethology and Interactive Creatures*. PhD thesis, Massachusetts Institute of Technology, Media Lab, 1996.
- [19] B Blumberg. Go with the flow: Synthetic vision for autonomous animated creatures. In W.L. Johnson and B. Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents*, pages 538–539, New York, 1997. ACM Press.
- [20] B Blumberg. (void*): A cast of characters. *Proceedings of SIGGRAPH 99: conference abstracts and applications*, 1999.
- [21] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M.P. Johnson, and B. Tomlinson. Integrated learning for synthetic characters. In *Proceedings of the 29th annual ACM conference on Computer Graphics and Interactive Techniques*, volume 21, pages 417–426. SIGGRAPH, 2002.
- [22] B. Blumberg and T. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of SIGGRAPH 1995*, Computer Graphics Proceedings, Annual Conference Series. ACM, ACM Press, 1995.
- [23] A.F. Bobick and Y.A. Ivanov. Action recognition using probabilistic parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 198–202, Santa Barabara, CA, 1998.
- [24] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Santa Fe, New Mexico, July 1999.
- [25] C. Breazeal. *Designing Sociable Robots*. MIT Press, 2002.
- [26] C. Breazeal. Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies*, 59:119–155, 2003.

- [27] C. Breazeal. Social interactions in hri: The robot view. *IEEE SMC Transactions, Part C.*, Forthcoming 2004.
- [28] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, and B. Blumberg. Learning from and about others: Towards using imitation to bootstrap social understanding in robots. *Artificial Life*, forthcoming, 2004.
- [29] C. Breazeal, G. Hoffman, and A. Lockerd. Teaching and working with robots as a collaboration. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*. AAMAS, July 2004.
- [30] C. Breazeal and B. Scassellati. Challenges in building robots that imitate people. In Dautenhahn and Nehaniv [42].
- [31] C. Breazeal and B. Scassellati. Robots that imitation humans. *Trends in Cognitive Sciences*, 6:481–487, 2002.
- [32] A. Brooks, J. Gray, A. Lockerd, H. Lee, and C. Breazeal. Robot’s play: Interactive games with sociable machines. In *Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. ACE 2004, June 2004.
- [33] R Burke. It’s about time: Temporal representations for synthetic characters. Master’s thesis, Massachusetts Institute of Technology, Media Lab, 2001.
- [34] R. Burke, D. Isla, M. Downie, Y. Ivanov, and B. Blumberg. Creature smarts: The art and architecture of a virtual brain. In *Proceedings of the 2001 Computer Game Developers Conference*, 2001.
- [35] R.W. Byrne and A.E. Russon. Learning by imitation: A hierarchical approach. In *Behavioral and Brain Sciences* [1], pages 667–684.
- [36] J. Call and M. Carpenter. Three sources of information in social learning. In Dautenhahn and Nehaniv [42], pages 211–228.

- [37] J. Cassell. Nudge nudge wink wink: Elements of face-to-face conversation in embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*. MIT Press, Cambridge, MA, 1999.
- [38] M. Cook and S. Mineka. Observational conditioning of fear to fear-relevant versus fear-irrelevant stimuli in rhesus monkeys. *Journal of Abnormal Psychology*, 98(4):448–459, 1989.
- [39] G. Csibra. Teological and referential understanding of action in infancy. *Philosophical Transactions of the Royal Society of London*, 358:447–458, February 2003.
- [40] E. Curio, U. Ernst, and W. Vieth. Cultural transmission of enemy recognition: One function of mobbing. *Science*, 202:899–901, 1978.
- [41] K. Dautenhahn. Getting to know each other - artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16:333–356, 1995.
- [42] K. Dautenhahn and C.L. Nehaniv, editors. *Imitation in animals and artifacts*. MIT Press, Cambridge, MA, 2002.
- [43] M. Davies and T. Stone. *Mental Simulation*. Blackwell Publishers, Oxford, 1995.
- [44] R. Dawkins. Hierarchical organization: A candidate principle for ethology. In P.P.G. Bateson and R.A. Hinde, editors, *Growing points in ethology*. Cambridge University Press, 1976.
- [45] F.B.M. de Waal. No imitation without identification. In *Behavioral and Brain Sciences* [1], page 689.
- [46] J. Demiris and G. Hayes. Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model. In Dautenhahn and Nehaniv [42].

- [47] M. Downie. behavior, animation and music: the music and movement of synthetic characters. Master's thesis, Massachusetts Institute of Technology, Media Lab, 2001.
- [48] D. Eberly. *3D Game Engine Design*. Morgan Kaufmann, 2001.
- [49] J. Fisher and R.A. Hinde. The opening of milk bottles by birds. *British Birds*, 42:347–357, 1950.
- [50] A. Fod, M.J. Mataric, and O.C. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, January 2002.
- [51] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42:143–166, 2003.
- [52] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119:593–609, 1996.
- [53] V. Gallese and A. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501, 1998.
- [54] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
- [55] M. Gleicher. retargetting motion to new characters. In *Proceedings of SIG-GRAPH 1998*, Computer Graphics Proceedings, Annual Conference Series, pages 33–42. ACM, ACM Press, 1998.
- [56] M. Gleicher, H.J. Shin, L. Kovar, and A. Jepsen. Snap together motion: Assembling run-time animation. In *Symposium on Interactive 3D Graphics*, April 2003.
- [57] R. Gordon. Folk psychology as simulation. *Mind and Language*, 1:158–171, 1993.
- [58] B. Hare, J. Call, B. Agnetta, and M. Tomasello. Chimpanzees know what conspecifics do and do not see. *Animal Behaviour*, 59:771–785, 2000.

- [59] B. Hare, J. Call, and M. Tomasello. Do chimpanzees know what conspecifics know? *Animal Behaviour*, 61:139–151, 2001.
- [60] G. Hayes and J. Demiris. A robot controller using learning by imitation. In *Proceedings of the Second International Symposium on Intelligent Robots and Systems*, pages 198–204, Grenoble, France, 1994. LIFTA-IMAG.
- [61] M. Heimann. When is imitation imitation and who has the right to imitate? In *Behavioral and Brain Sciences* [1], page 693.
- [62] C.M. Heyes and B.G. Galef, editors. *Social Learning in Animals: The Roots of Culture*. Academic Press, Boston, 1996.
- [63] G. Hoveland, P. Sikka, and B. McCarragher. Skill acquisition from human demonstration using a hidden markov model. In *IEEE International Conference on Robotics and Automation*, pages 2706–2711. ICRA '96, IEEE, 1996.
- [64] M.A. Huffman. Acquisition of innovative cultural behaviors in nonhuman primates: A case study of stone handling, a socially transmitted behavior in japanese macaques. In Heyes and Galef [62], pages 267–286.
- [65] D. Isla. The virtual hippocampus: Spatial common sense for synthetic creatures. Master's thesis, Massachusetts Institute of Technology, Media Lab, 2001.
- [66] D. Isla and B. Blumberg. Object persistence for synthetic characters. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*. AAMAS, 2002.
- [67] O.C. Jenkins and M. Mataric. Primitive-based movement classification for humanoid imitation. Technical Report IRIS-00-385, University of Southern California, Institute for Robotics and Intelligent Systems, 2000.
- [68] M.P. Johnson, A. Wilson, B. Blumberg, C. Kline, and A. Bobick. Sympathetic interfaces: Using a plush toy to direct synthetic characters. In *Proceedings of the CHI '99 Conference on Human Factors in Computing Systems*, 1999.

- [69] D. Keltner. Social functions of emotions at four levels of analysis. *Cognition and Emotion*, 13:505–521, 1999.
- [70] H. Kozima. Attention-sharing and behavior-sharing in human-robot communication. In *IEEE International Workshop on Robot and Human Communication*, pages 9–14, Takamatsu, Japan, 1998. ROMAN '98.
- [71] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge form visual observation of human performance. *IEEE Transactions on Robotics Automation*, 10:799–822, 1994.
- [72] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
- [73] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of the 29th annual ACM conference on Computer Graphics and Interactive Techniques*. SIGGRAPH, 2002.
- [74] J.M. Leppanen. Does emotion recognition accuracy predict social competence in childhood? *Psychologia*, 36:429–438, 2001.
- [75] J. Lieberman. Teaching a robot manipulation skills through demonstration. Master's thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, June 2004.
- [76] A. Lockerd and C. Breazeal. Tutelage in socially guided robot learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IROS 2004, forthcoming 2004.
- [77] M Mataric. Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics. In Dautenhahn and Nehaniv [42], pages 391–422.

- [78] M. Mataric, V.B. Zordan, and M.M. Williamson. Making complex articulated agents dance; an analysis of control methods drawn from robotics, animation, and biology. *Autonomous Agents and Multi-Agent Systems*, 2(1):23–44, March 1999.
- [79] J.M. Mateo and W.G. Holmes. Development of alarm-call responses in belding’s ground squirrels: the role of dams. *Animal Behaviour*, 54:509–524, 1997.
- [80] M.D. Matheson and D.M. Fragaszy. Imitation is not the “holy grail” of comparative cognition. In *Behavioral and Brain Sciences* [1], pages 697–698.
- [81] A. Meltzoff. Infant imitation after a 1-week delay: Long-term memory for novel acts and multiple stimuli. *Developmental Psychology*, 24:470–476, 1988.
- [82] A. Meltzoff and M.K. Moore. Imitation, memory, and the representation of persons. *Infant behavior and Development*, 17:83–99, 1994.
- [83] A.N. Meltzoff. Understanding the intentions of others: Re-enactment of intended acts by 18-month-old children. *Developmental Psychology*, 31:838–850, 1995.
- [84] A.N. Meltzoff. The human infant as imitative generalist: A 20-year progress report on infant imitation with implications for comparative psychology. In Heyes and Galef [62], pages 347–370.
- [85] A.N. Meltzoff and J. Decety. What imitation tells us about social cognition: a rapprochement between developmental psychology and cognitive neuroscience. *Transactions of the Royal Society of London B*, 358:491–500, 2003.
- [86] A.N. Meltzoff and A. Gopnik. The role of imitation in understanding persons and developing a theory of mind. *Developmental Psychology*, 24:470–476, 1993.
- [87] A.N. Meltzoff and M.K. Moore. Explaining facial imitation: A theoretical model. *Early Development and Parenting*, 6:179–192, 1997.

- [88] H. Miyamoto and M. Kawato. A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks*, 11:1131–1344, 1998.
- [89] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato. A kendama learning robot based on bi-directional theory. *Neural Networks*, 9:1181–1302, 1996.
- [90] E. Oztop and M.A. Arbib. Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics*, 87(2):116–140, 2002.
- [91] K. Perlin and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. In *Proceedings of SIGGRAPH 1996*, Computer Graphics Proceedings, Annual Conference Series. ACM, ACM Press, 1996.
- [92] R.W. Picard. *Affective Computing*. MIT Press, Cambridge, 1997.
- [93] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, 1978.
- [94] W. Prinz. A common coding approach to perception and action. In O. Neumann and W. Prinz, editors, *Relationships between perception and action*, pages 167–201. Springer-Verlag, Berlin, 1990.
- [95] R.P.N. Rao, A.P. Shon, and A.N. Meltzoff. A bayesian model of imitation in infants and robots. In K. Dautenhahn and C. Nehaniv, editors, *Imitation and Social Learning in Robots, Humans , and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge University Press, Forthcoming 2004.
- [96] M. Resnick. *Turtles, Termites and Traffic Jams*. MIT Press, Cambridge, MA, 1994.
- [97] C. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Proceedings of SIGGRAPH 1987*, Computer Graphics Proceedings, Annual Conference Series, Orlando, July 1987. ACM, ACM Press.

- [98] G. Rizzolati, L. Fadiga, V. Gallese, and L. Foggassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.
- [99] H.L. Roitblat. Mechanisms of imitation: The relabeled story. In *Behavioral and Brain Sciences* [1], pages 701–702.
- [100] D. Roy, K. Hsiao, and N. Mavridis. Conversational robots: Building blocks for grounding word meanings. In *Proceedings of the HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data*, 2003.
- [101] D. Roy, K. Hsiao, and N. Mavridis. Mental imagery for a conversational robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3):1374–1383, 2004.
- [102] B. Scassellati. Theory of mind for a humanoid robot. In *Proceedings of the International Conference on Humanoid Robotics*. IEEE/RSJ, September 2000.
- [103] S. Schaal. Learning from demonstration. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 1040–1046. MIT Press, Cambridge, MA, 1997.
- [104] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3:233–242, 1999.
- [105] R.M. Seyfarth, D.L. Cheney, and P. Marler. Vervet monkey alarm calls: Semantic communication in a free-ranging primate. *Animal Behaviour*, 28:1070–1094, 1980.
- [106] P. Slater. Bird song learning: causes and consequences. *Ethology, Ecology, and Evolution*, 1:19–46, 1989.
- [107] F. Strack, L. Martin, and S. Stepper. Inhibiting and facilitating conditions of the human smile: A nonobtrusive test of the facial feedback hypothesis. *Journal of Personality and Social Psychology*, 54:768–777, 1988.
- [108] J. Terkel. Cultural transmission of feeding behavior in the black rat (*rattus rattus*). In Heyes and Galef [62], pages 17–45.

- [109] W. Timberlake. Motivational modes in behavior systems. In R.R. Mowrer and S.B. Klein, editors, *Handbook of contemporary learning theories*. Erlbaum Associates, Hillsdale, NJ, 2002.
- [110] W. Timberlake and G.A. Lucas. Behavior systems and learning: from misbehavior to general principles. In R.R. Mowrer and S.B. Klein, editors, *Contemporary learning theories: Instrumental conditioning theory and the impact of biological constraints on learning.*, pages 237–275. Erlbaum Associates, Hillsdale, NJ, 1989.
- [111] N. Tinbergen. *The Study of Instinct*. Oxford University Press, 1951.
- [112] M. Tomasello, J. Call, and B. Hare. Five primate species follow the visual gaze of conspecifics. *Animal Behaviour*, 55:1063–1069, 1998.
- [113] M. Tomasello, J. Call, and B. Hare. Chimpanzees understand psychological states—the question is which ones and to what extent. *Trends in Cognitive Science*, 7:153–156, 2003.
- [114] M. Tomasello, M. Davis-Dasilva, L. Camak, and K. Bard. Observational learning of tool use in young chimpanzees. *Human Evolution*, 2:175–183, 1987.
- [115] M. Tomasello, B. Hare, and B. Agnetta. Chimpanzees, *Pan troglodytes*, follow gaze direction geometrically. *Animal Behaviour*, 58:769–777, 1999.
- [116] B. Tomlinson, M. Downie, M. Berlin, J. Gray, D. Lyons, J. Cochran, and B. Blumberg. Leashing the alphawolves: Mixing user direction with autonomous emotion in a pack of semi-autonomous virtual characters. In *Proceedings of the Symposium on Computer Animation*, 2002.
- [117] X. Tu and D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of SIGGRAPH 1994*, Computer Graphics Proceedings, Annual Conference Series, Orlando, July 1994. ACM, ACM Press.

- [118] S. Weber, M. Mataric, and O.C. Jenkins. Experiments in imitation using perceptuo-motor primitives. *Autonomous Agents*, pages 136–137, 2000.
- [119] A. Whiten. Studies of imitation in chimpanzees and children. In Heyes and Galef [62], pages 291–318.
- [120] A. Whiten. When does smart behaviour-reading become mind reading? In P. Carruthers and P. Smith, editors, *Theories of Theories of Mind*. Cambridge University Press, 1996.
- [121] A. Whiten. Imitation of sequential and hierarchical structure in action: experimental studies with children and chimpanzees. In Dautenhahn and Nehaniv [42], pages 191–209.
- [122] A. Whiten. The dissection of socially mediated learning. forthcoming 2004.
- [123] A. Whiten and W. Byrne. *Machiavellian Intelligence II: Extensions and Evaluations*. Cambridge University Press, 1997.
- [124] J.H. Willams, A. Whiten, T. Suddendorf, and D.I. Perrett. Imitation, mirror neurons and autism. *Neuroscience and Biobehavioral Review*, 25(4):285–295, June 2001.
- [125] A. Wilson and A. Bobick. Realtime online adaptive gesture recognition. In *Proceedings of the International Conference on Pattern Recognition*, Barcelona, Spain, September 2000.
- [126] D. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.
- [127] Y. Wu and S.T. Huang. Vision-based gesture recognition: A review. In A. Braffort, editor, *Lecture Notes in Artificial Intelligence 1739, Gesture-Based Communication in Human-Computer Interaction*, Gif-sur-Yvette, France, 1999. International Gesture Workshop, GW '99.