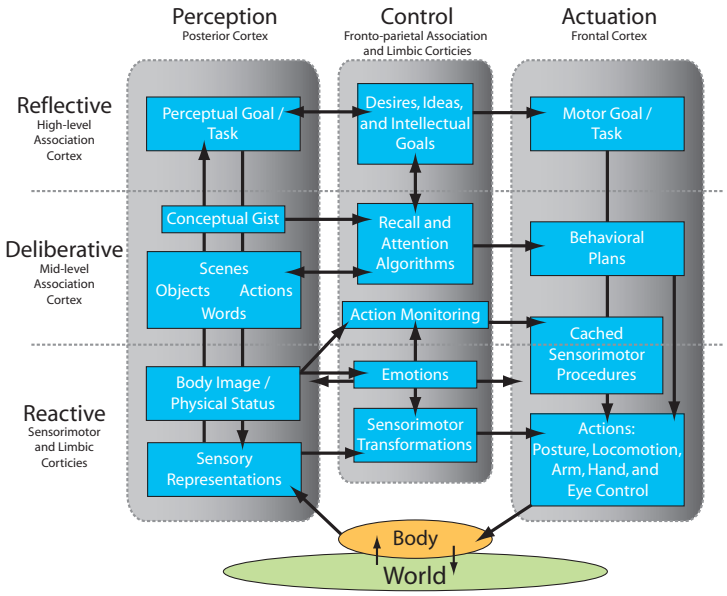
CHIP: A Cognitive Architecture for Comprehensive Human Intelligence and Performance

A Report of the CHIP Project:
Howard E. Shrobe and Patrick H. Winston – Principal Investigators
MIT Computer Science and Artificial Intelligence Laboratory
MIT Brain and Cognitive Science Department
MIT Media Laboratory
SRI International
Date: September 29, 2006 10: 0 8 P.M.

Authors:

Howard Shrobe	MIT CSAIL	Patrick Winston	MIT CSAIL
Josh Tennenbaum	MIT BCS	Pat Shaftoe	MIT BCS
Steve Massaquoi	MIT Health Science and Technology	Paul Robertson	MIT CSAIL
Brian Williams	MIT Aero and Astronautics Dept.	Ian Eslick	MIT Media Laboratory
Sajit Rao	MIT CSAIL	Michael Coen	MIT CSAIL
Antonio Torralba	MIT CSAIL	William Freeman	MIT CSAIL
Rusty Bobrow	BBN		



Contents

Chapter 1

An architecture for human cognition

1.1 A scenario

George is about to pay his bills and is trying to find his pen. He is attached to it because he's had it for a long time. He can picture putting it down on the kitchen table, but he doesn't see it there. He thinks about where else he usually puts the pen and decides to look in the living room first because it's closer than the study; but he doesn't see it there. Walking back to the kitchen, he thinks he sees a shiny object in a cup holding odds and ends. As he turns to get a better look at it, he hears a noise and jerks around towards it; he sees the cat running away. He turns back towards the cup, spills its contents on the table and sees that the shiny object is in fact his pen.

1.2 What the scenario teaches us

People are complicated, even in the most mundane situations. We think at many different levels: we react, we deliberate, and we reflect on what we've done. We are situated in and react to our environments while we simultaneously pursue our goals and find ways to translate them into actions. We monitor our actions to make sure that they have the effect we intended. We have complicated internal state, including memories, emotions, desires, a sense of place and task context. We know an incredible amount about the everyday world about us and manage the complexity of the world by appropriately abstracting away details not relevant to the circumstances. We use our imaginations to see what might happen were we to take an action, in effect using our motor and perceptual systems to think.

1.3 What biology teaches us

Research during the past century, and expecially during the past few decades, has filled vast libraries with insights into the organization of our brains and the mechanisms at work in them. From the BICA perspective, the most important of those insights include the following:

- **Cognition is highly parallel, occurs at many interconnected levels**, and can be best understood in terms of the interactions between a large and diverse collection of interacting sub-systems. Within each of the cerebral cortical areas, basal ganglia, cerebellum, brainstem-spinal cord there appear to be systems that are simpler, and phylogenetically older, that still handle core operations in the adult human, as well as other systems that are phylogenetically newer that manage our most advanced capacities. There tends to be extensive retained interaction among the old components, and there is rich interconnection with the new components. As a result, a significant onionlike structure appears to exist in the CNS (Central Nervous System), with outer layer functions designed to exploit and build upon those that can be relegated to lower systems lying deeper in the brain.

- **Cognition involves both bottom up and top down processing**. More importantly, cognition involves tightly integrated loops, with both bottom-up and top-down elements, that constantly influence one another. For motor control, negative feedback is a powerful general method for achieving goals and positive feedback is a powerful general method for performing constrained optimization. For perception, top-down contextual biases from an object model or the gist of the scene modify and tune the responses of lower-level neurons.

- **Memory structures are multi-faceted**. Memory appears to exist in distributed form such that storage and use are collocated. There is little if any transport of stored information. However, there also seem to be distinct memory systems: those that hold semantic knowledge, those that hold remembered experiences, those that hold procedural knowledge, and those that hold perceptual memories.

- **There are many ways of learning:** we can learn a concept by being shown examples and counter-examples; we can learn by being instructed verbally; we can learn a motor skill by practicing it; we can learn how to do something by observing someone else doing it; we can learn how to do a complex task by caching and generalizing a plan that we've assembled for that task; and we can learn by making analogies between a current task and some previous experience.

These are the special insights from biology that have inspired the cognitive architecture that we present in the following sections.

1.4 Core architectural principles

The *Comprehensive Human Intelligence Project* (CHIP) architecture, is an evolving framework for building a system that attempts to encompass the full range and magic of human cognition. A high level view of the CHIP architecture is shown in figure 1.1. This is, of course, an abstract view of CHIP in which detail is suppressed. It is also only one among several ways of describing the CHIP architecture.

As shown in the figure, there are two principle dimensions along which we can organize CHIP's components and their interactions. Along the horizontal dimension we divide the system into components that deal with perception, actuation and control/decision-making. Along the vertical dimension we divide the components that operate at a reactive level, a deliberative level, and a reflective level, including self-awareness and social awareness. Each box of this matrix does not represents a single component, but rather a complex ensemble of components with significant internal complexity. Many of the features that one might expect to see at the top level of a cognitive architecture (e.g. short term memory) appear in CHIP as features of one of these sub-systems.

In subsequent chapters we will describe in more detail the component representations and processes of the sub-systems. Here, however, we emphasize that the architecture emerges from **five core architectural principles** that capture the essence of human cognition:

- Ubiquitous learning
- Tightly coupled, interacting, top-to-bottom loops
- Re-use of perceptual and motor mechanisms for abstract reasoning
- Combinators, closely aligned with language, allow unlimited composition of multifaceted concepts into complex descriptions
- Adaptive decision making and the influence of emotions

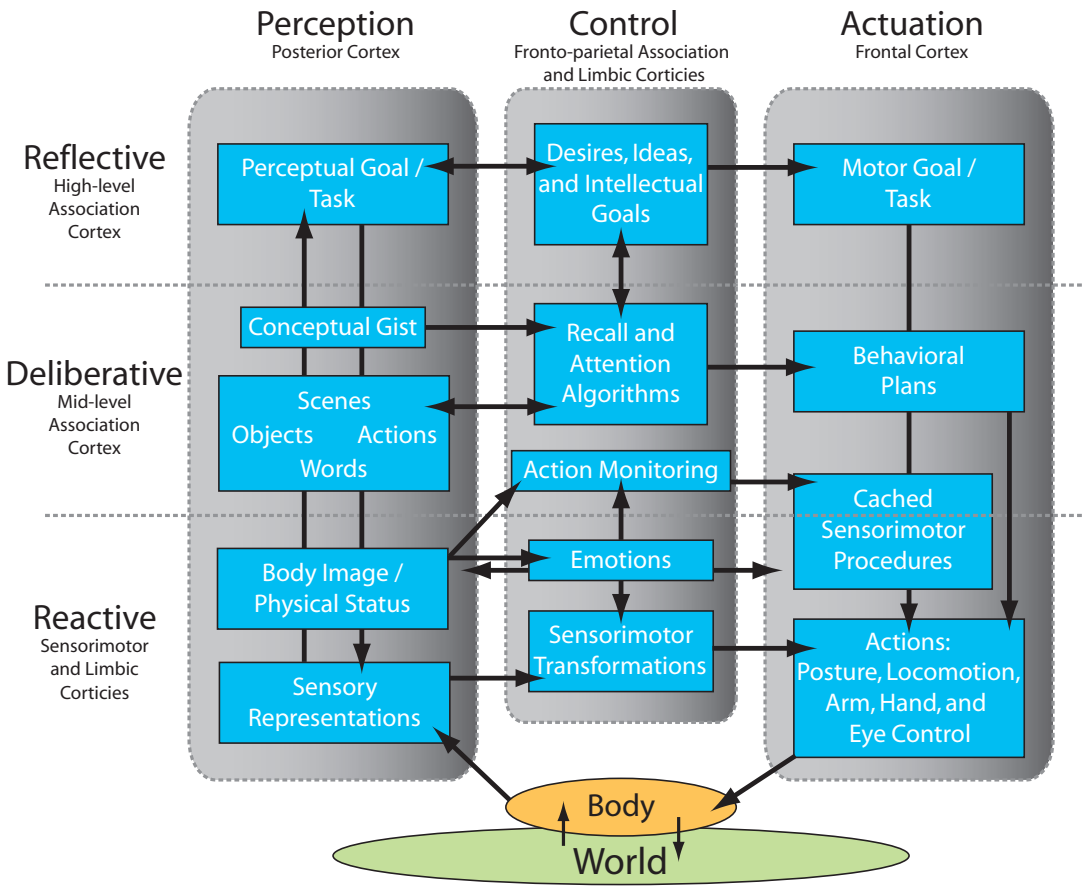


Figure 1.1: The CHIP Architecture: the systems of components intimately interact with one another, forming tightly integrated loops through the various layers. The middle column, which manages the internal state of the system, acts as a shared blackboard [51, 33, 10, 17] with three levels for general communication among the components. But in addition to this shared channel, there are many special purpose channels linking specific sub-systems. The figure also maps this general organization onto the gross anatomy and functionality of the human brain.

1.4.1 Ubiquitous learning

Learning is everywhere in human cognition, so learning is everywhere in the CHIP architecture. Learning takes place in tightly coupled loops; learning is enabled by perceptual reuse; every combinator-enabled representation enables learning, and emotions guide learning. Because the system has a reflective layer that looks at the system's internal state, it can learn about its own thinking using the same mechanisms that help it learn about the world. Learning, therefore, not only helps a CHIP-based system to improve its performance, learning helps it improve its thinking.

Section 2 of this report focuses directly on learning; but it should be remembered that every other section tells a learning story as well.

1.4.2 Tightly coupled, interacting, top-to-bottom loops

CHIP is neither a strictly top down, nor bottom up architecture. It necessarily involves highly complex and emergent patterns of interaction between its components. One useful way to look at this architectural diagram is to trace paths through it.

All behavior is effected by loops that tightly couple perceptual representations, decision-making, or control machinery to action. The loops are at different hierarchical levels (reactive, deliberative, and reflective) with loops at higher levels modifying the behavior of lower ones. Consider, for example, what happens when George sees a moving object and turns towards it. This involves coupled perception and action without any conscious effort; it is a reactive process. However, George could also have also noticed the motion, figured it was just the cat and decided to ignore it; this would be a deliberative process. Both of these are driven bottom up and originate with perception.

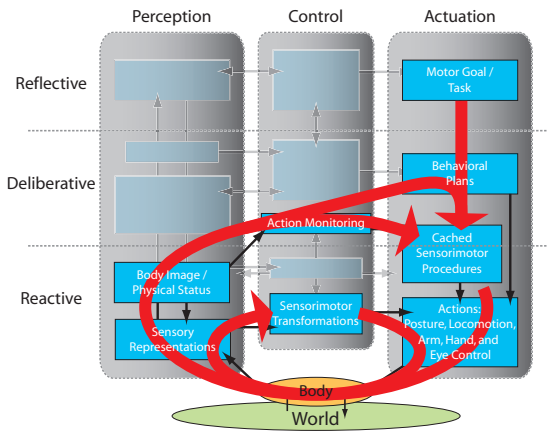
Many other processes such as the actuation process shown in figure 1.2(a) start with internal goals and involve complex reasoning about how to achieve the plans; such an activity is a top-down, deliberative action. Once a plan of action is formed, it usually results in (among other things) the execution of previously learned motor activity that proceeds automatically and in real-time; but even this activity is monitored by internal sensory systems that make sure we're not tripping over our own shoelaces. This control loop is driven top down, but executed at the reactive level as a sensory-motor routine. For example, it is the reactive layer that controls the placement of the foot while walking; it also produces the control actions that compensate for minor variations in placement or terrain. However, a larger loop leads back to a more deliberative level of self-monitoring that is invoked if the lower level is perturbed beyond the envelope controllable by the reactive level; for example, consider walking across a balance beam and missing the beam. In such cases, the qualitative shape of the plan must be changed, or a different plan selected.

The loops mentioned are *intermodule* because they go across modules, connecting perception to control structures to action. There are also tightly coupled loops that are *intramodule*, that is, within the perception or actuation columns of Figure 1.1. In perception, for instance, feed-forward streams rapidly convey bottom-up information from the sensors, while context and the task send top-down, feedback signals that serve to fine-tune, filter, organize, and interpret the bottom-up signal.

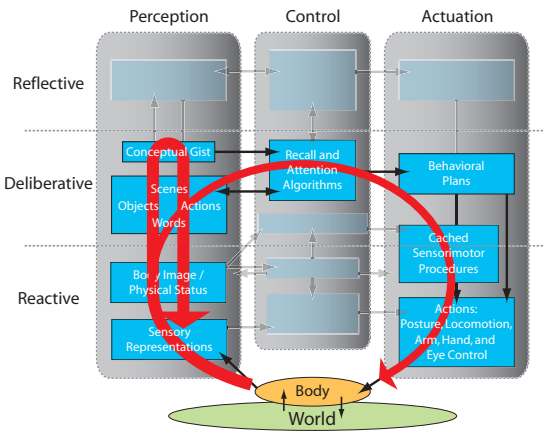
As an example, consider the perceptual processing involved in recognizing an object that has come into view. This process begins bottom up; there is strong evidence that the human visual system contains a fast pathway that uses coarse features to get a "gist" of a scene, to identify, for example, whether one is in an office or a market place. The gist then provides strong top-down biases for where certain types of objects should be found (for example, in an office one would look for a bookcase on the wall). These expectations drive across at the deliberative level, posting a goal to get a better view of those objects. This is refined, top-down, into a set of motor routines that either saccade the eyes or move the body (or both). At the same time the expectations drive top-down in the perceptual system, setting a set of biases for what objects should be recognized. The loops involved in this process are shown in figure 1.2(b).

1.4.3 Re-use of perceptual and motor mechanisms for abstract reasoning

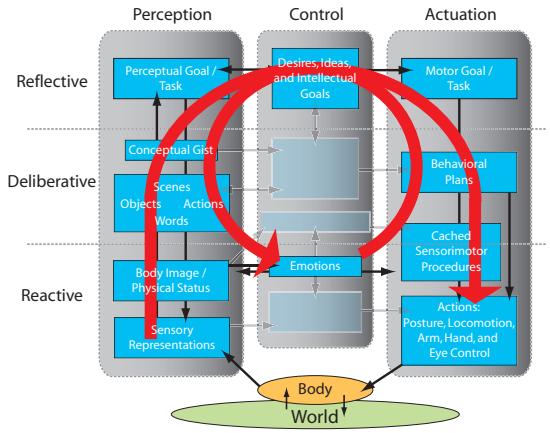
Sensorimotor representations and processes are important components of cognition because we think with our eyes, our mouths, and our hands and legs. They are not merely "bolted-on" ornaments that deliver symbols to a core



(a) Loops involved in motor control



(b) Loops involved in scene understanding



(c) Loops involved in emotion influencing goals

Figure 1.2: The dynamics of tightly coupled loops leads to complex behavior

reasoning system. Instead, they are faculties that individually and collectively enable learning and reasoning. For example, if we want to learn about what something looks like, we know we have to look at it from multiple points of view, and we can arrange to see that thing from those multiple points of view by rotating it with our hands or walking around it. Thus, we learn about the world and solve problems by looking at, touching, and moving things. In many cases, we do not even have to be in the world at the time, for our perception and motor apparatus enables us to imagine how things behave in the physical world, and to read answers off of our imagined world.

1.4.4 Combinators, closely aligned with language, allow unlimited composition of multifaceted concepts into complex descriptions

Whenever we think, we tie concepts together. If George thinks about looking for his pen, he combines the pen concept with a search procedure, producing a looking-for-pen concept. If he thinks about putting the pen on the kitchen table, he combines the pen concept with a trajectory concept terminating at the table, producing a moving-pen concept.

When George talks, his words fall into structures that amount to recipies for combining concepts in the brain of those who are listening. When George talks to himself, the structures seem to provide access to memories of conceptually similar combinations, thus providing the foundation for analogical problem solving.

Thus, language is not just a medium of communication, it is window into a uniquely human ability to combine

concepts into new concepts. We call this the **combinator** capability. Mounting evidence suggests that humans possess the combinator capability at a level that other animals cannot begin to match. It follows that other animals cannot have a human-level language, because human-level language requires the assembly of words into combinations without practical limit, and other animals also cannot have anything like the human capability for analogical problem solving, for analogical problem solving likewise requires the assembly of all sorts of perceptually grounded concepts—such as events, causes, changes, animate and inanimate objects— into complex, layered, story-like descriptions.

Also, there is a marshalling dimension to combinators and language, and saying things to ourselves may be even more important than saying things to others. Anyone told that "the vehicles collided" and asked "did the vehicles touch each other" says "yes" and reports that that it must be true because he imagines the scene and read the contact off of the imagined scene as if it were real. Thus, language has an essential role in stimulating the reuse of visual perceptions to deal with imagined worlds. And the reuse of perceptions causes new thoughts to arise, expressed in language, which causes new reuse of perception, constituting yet another tightly coupled loop.

1.4.5 Adaptive decision making and the influence of emotions

Most models of human cognition have focused on rational cognition and have substantially ignored the critical issues of emotion and motivation. Our architecture involves mechanisms for adaptive decision making that use decision theoretic techniques to make choices that are rational with respect to the systems' beliefs and goals. However, it also recognizes that emotions at the reactive level have a substantial influence on behavior by affecting higher-level desires and intentions as well as changing the evaluation criteria for ongoing action.

Consider what would happen if the cat in our scenario actually broke George's pen. George might become really angry. As shown in the digram in figure 1.2(c), the bottom up processing of the imagery of the cat goes from visual signal, to scene understanding and object recognition, to event recognition. The recognition of the event in which the cat breaks the pen triggers the emotional reaction of getting angry. Decision making is strongly influenced by emotional state (as well as by other aspects of context). A change in emotional state results in changes in the way we evaluate which goals are important; it also results in changes in the way we evaluate which methods for achieving these goals are more desirable. In the CHIP architecture, this is realized by the activation of a set of decision making agents different from those operating in more normal states. This results in behavior that is quite different from that which is normal. Continuing to follow the process, the change in emotional state, allows the goal of killing the cat to be activated and translated into plans and actions. George goes to get his gun.

1.5 Overview of the rest of the document

1.5.1 Learning and memory

Figure 1.3 shows the variety of memories and learning techniques that are used in the CHIP architecture. At the low level there are memory systems that hold the parameters that control the basic operations of perception and actuation. These parameter memories are populated by a novel form of self-supervised learning called "multi-modal clustering".

As described in 2, procedural memory holds actuation procedured, cached hierarchical sequences of basic operations, that are executed as a unit at performance speed. These are acquired through a compilation process that monitors deliberative execution of these actions using Hebbian learning mechanisms. At the Higher levels CHIP maintains an episodic and semantic memory and the associated perceptual memories that hold imagery, sounds, etc. A variety of Bayesian based learning techniques are used to learn the information held in these memories and to extract more general information from them. In addition, one shot learning techniques, such as forming macro-operators [23], chunking, [38], explanation-based learning [46] and learning by debugging [79] are performed as a side-effect of reasoning and decision making.

1.5.2 Actuation

In Chapter 4 we turn to the problem of actuation. In the context of the CHIP architecture, actuation is viewed as a loop from perception to actuation. The actuation loop is described in relation to Basal Ganglionic circuits in

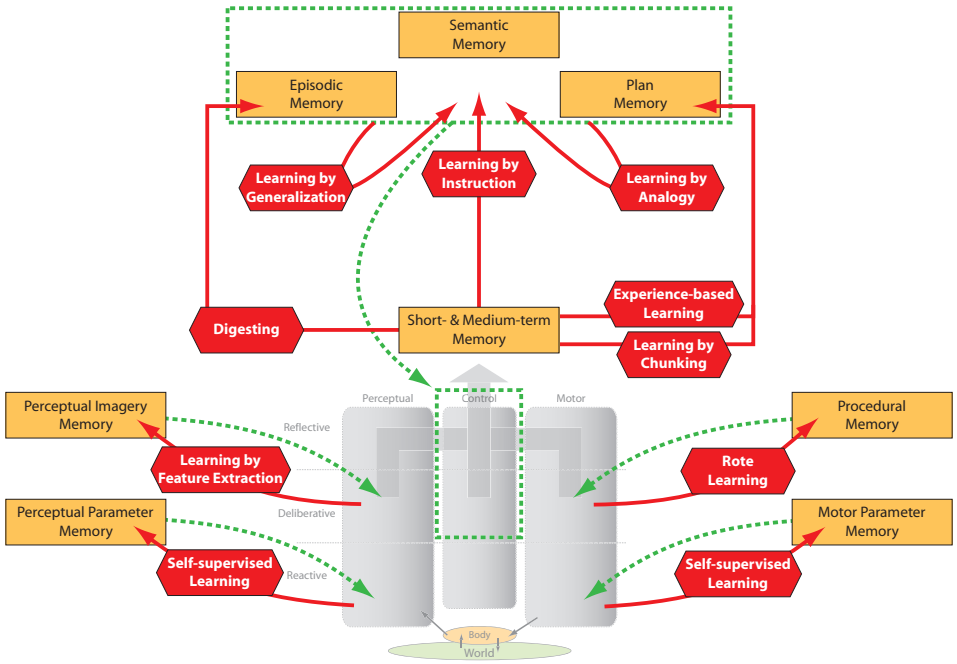


Figure 1.3: Memory systems and corresponding learning pathways in the CHIP architecture.

the brain. It is shown how complex patterns of actuation can be achieved through cached sensorimotor procedures and a qualitative state plan (QSP). While moderate sized implementations of the actuation loop that we describe are possible using conventional hardware we discuss how dramatically larger systems can be built using industry standard components (FPGAs) that can implement high fan-in circuits.

1.5.3 Perception

As described in 3, we explain why we deviate significantly from traditional systems in which perception is dismissed as a "front-end" module whose job is only to deliver symbols about the state of the world to a symbolic reasoning system. Instead, in our architecture, non-symbolic perceptual representations and processes are central to reasoning. Key aspects of our architecture for perception are:

- Three qualitatively different kinds of perceptual representations and processes.
- Independent, hierarchical loops that tightly couple perception to action.
- Interaction of bottom-up and top-down processes.
- Re-use of perceptual representations and processes for reasoning.

To illustrate these key features of our architecture, we present a sample of results from work already done.

Learning object and scene models and top-down influences of scene context: We show how a system learns about the *gist* or *context* of a scene and how it uses that representation to inform it about where to look for specific objects or feature of the scene.

Taskability—Goal or task driven perception: We show how a system uses a sequence of primitive visuospatial operations to locate what a person might be pointing at. The *visual routine* for pointing is just one of many visual routines that can be constructed from the same set of base primitives.

Active, multi-modal, object recognition: we show how a robot manually explores graspable objects and builds a representation that combines visual and proprioceptive information, which it subsequently uses for recognizing the object.

1.5.4 Decision making

The CHIP decision-making architecture is shown in figure 1.4. As we explain in 5, it is a multi-tiered, adaptive problem solving system, that is based on preference driven, decision-theoretic techniques. All of the agents that contribute to decision-making are sensitive to context and emotional state, meaning that changes in context, or changes in the system's emotional state, lead to changes in its view of what goals are important to accomplish as well changes in its view of what are attractive ways of accomplishing these goals. Plan execution is accomplished partly deliberatively and partly procedurally through interaction with the actuation system and is monitored through interaction with the perceptual system. Breakdowns are diagnosed and the diagnosis is used to select repair strategies.

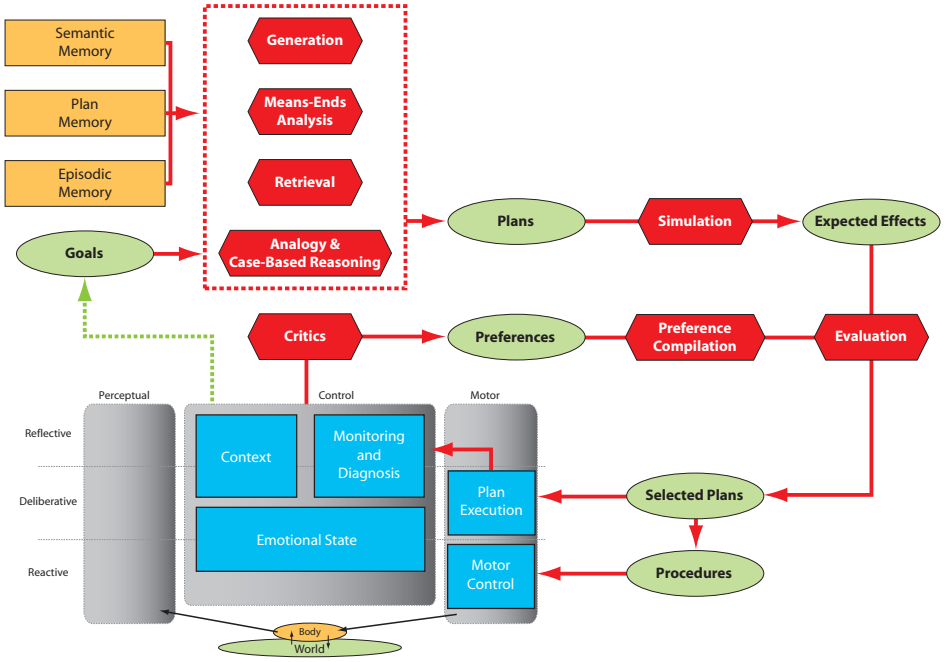


Figure 1.4: The CHIP Decision Making Architecture

This overall structure leads to the following properties of the CHIP architecture:

- It can find more than one way to do almost everything.
- It is able to pursue multiple goals simultaneously.
- It is able to prioritize among competing goals.
- It is able to adapt to changing circumstances.
- It is sensitive to changes in context.
- It is influenced by its current emotional state.
- It is able to monitor its own actions and intervene when things go wrong.
- It is able to operate reflectively, applying its planning and decision making capabilities to its own thinking.

1.5.5 Representation and Language

In 6, we argue that the ability to combine two concepts into a third concept without limit and without destroying the constituents is uniquely human and closely tied to human linguistic competance. The combinator capability is a prerequisite to the construction of any representation, and the construction of representations lies at the heart of building models, which is a prerequisite to understanding the present, explaining the past, predicting the future, and controlling the future.

Also, language is closely aligned with two other capabilities of central importance:

- Recording and accululating experiences that shed light on new problems.
- Actuating the reuse of perceptual and motor faculties such that imagination can participate in problem solving.

1.5.6 Novel aspects of the chip architecture

The CHIP architecture is build from a collection of interacting pipelines, each of which involves the counterflow of top-down and bottom-up information and each maintains cross connections with other pipelines. We describe these pipelines, called *Influence Networks* in more detail in Chapter 2. The control and decision making pipeline maintains a multi-level blackboard with reactive, deliberative, and reflective layers.

The CHIP architecture is an integrated cognitive architecture that is novel in many respects:

- It recognizes multiple layers of cognitive processing
- It recognizes the integration of perception, actuation and decision making.
- It recognizes the critical role of adaptive decision making, while incorporating the first class role of the emotions in influencing decision making, perception and actuation.
- It recognizes a diversity of representations and mechanisms for populating them.
- It recognizes a diversity of memory systems and learning mechanisms for populating them.

Chapter 2

Learning and Memory

2.1 Introduction

The goal of a human-level learner is to take complex, noisy information from multiple modalities and distill this experience into a representation that supports prediction about and manipulation of the world. From the perspective of artificial intelligence, recognizing episodic structure in experience, reasoning about whether a novel animal is ferocious, discovering abstract kinds such as symptoms and diseases, and going to the store are highly non-trivial achievements, yet small children make these inferences robustly and with relative ease. The goal of the CHIP architecture is to provide a framework for interacting with the world such that these competencies may be achieved.

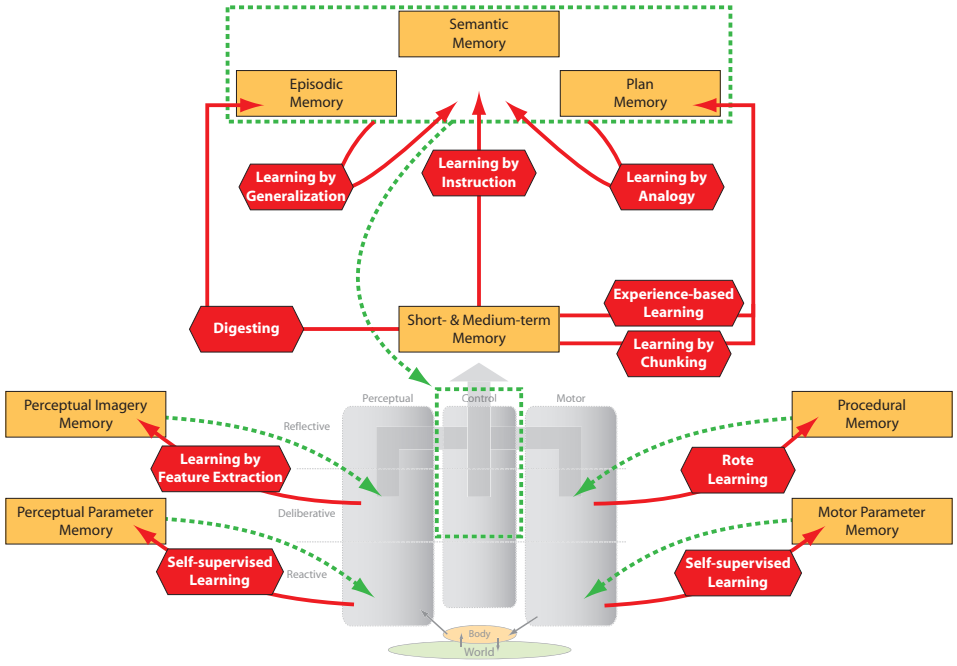


Figure 2.1: Memory systems and corresponding learning pathways in the CHIP architecture.

Figure 2.1 shows the variety of memory systems and learning mechanisms that are part of the CHIP architecture. At the lower level of the diagram are the memories that hold the parameters that guide perception and actuation; it is these memories that contain the information that allow us to recognize the phonemes and vowels of our native language or that allow us to place our limbs in a useful posture. These are acquired through self-supervised learning techniques that will be described next. Procedural memory holds compiled procedures that allow us to perform sequences of basic actions at performance speeds; this information is acquired through a compilation process that nevertheless preserves enough visibility into the execution to allow for adaptive control. The rest of the memories contain higher level information; the learning methods that populate these memories will be describe after we explain

how the contents of parameter memory are learned.

2.2 Self-supervised perceptual and sensorimotor learning

Young animals and children solve an array of challenging learning tasks. What makes these problems intriguing from an artificial intelligence standpoint is that their learning is largely unsupervised, particularly at early stages of development.

Consider, for example, a human infant who learns the phonetic structure of his native tongue simply through exposure to it. This is extraordinary because *a priori*, infants neither know the number of phonemes – which varies individually by language from 10 to approximately 140 – and they have no knowledge of the parameters governing the generation of the spoken language to which they are exposed.

In formal terms, we call this type of learning *non-parametric* and *distribution free*, in that it makes no assumptions about the number of different categories of data being presented or how the training data are probabilistically sampled from these categories. One may contrast this with standard mathematical approaches to clustering, where some knowledge of the clusters, e.g., how many there are or their statistical distributions, must be presumed in order to learn them. Without knowing these parameters in advance, many algorithmic clustering techniques tend not be robust [77].

Nonetheless, young animals (including humans) routinely solve these types of unsupervised learning problems in a wide variety of perceptual, sensorimotor, and cognitive domains. This suggests that *biological systems are capitalizing on some property of the world that most artificial, engineered systems have yet to take advantage of.*

Our recent work [7, 8, 9] suggests that this type of unsupervised learning is possible in part because of the tight correlations that exist between seemingly independent sensory modalities. The brain and cognitive sciences have gathered a enormous body of evidence demonstrating the extraordinary degree of interaction between sensory modalities during the course of ordinary perception (e.g., [6]. We present a framework for artificial perceptual systems that draws on these findings, where the primary architectural motif is the cross-modal transmission of perceptual information to structure and enhance sensory channels individually. In this section, we present self-supervised algorithms for learning:

- **Perceptual grounding**, which answers the first question that any natural (or artificial) creature faces: *what different things in the world am I capable of sensing?* This question is deceptively simple because a formal notion of what makes things different (or the same) is non-trivial and often elusive. We will show that animals (and machines) can learn their perceptual repertoires by simultaneously correlating information from their different senses, even when they have no advance knowledge of what events these senses are individually capable of perceiving.

  As a demonstration of this, we present a system that learns the number (and formant structure) of vowels in American English, simply by watching and listening to someone speak and then cross-modally clustering the accumulated auditory and visual data. The system has no advance knowledge of these vowels and receives no information outside of its sensory channels. This work is the first unsupervised acquisition of phonetic structure of which we are aware, at least outside of that done by human infants, who solve this problem easily.

- **Intersensory influence**, which shows how different sensory systems can mutually agree upon a common model of the world, even though these systems may receive noisy, even contradictory inputs. This influence allows to answer the question: once an animal (or a machine) has learned the range of events it can detect in the world, *how does it know what it's perceiving at any given moment?*

- **Sensorimotor learning** combines the above two components to enable learning sensorimotor control. This is surprising because one might suppose that motor activity is fundamentally different than perception. However, we take the perspective that motor control can be acquired through recursive, internal perception. The power of this mechanism is that it can learn mimicry, an essential form of behavioral learning (see the developmental sections of Meltzoff and Prinz 2002) where one animal acquires the ability to imitate some aspect of another's

activity, constrained by the capabilities and dynamics of its own sensory and motor systems. We will demonstrate sensorimotor learning in our framework with an artificial system that learns to sing like a zebra finch by first listening to a real bird sing and then by learning from its own initially uninformed attempts to mimic it.

2.2.1 Setting the stage

Our example begins with the 1939 World's Fair in New York, where Gordon Peterson and Harold Barney [54] collected samples of 76 speakers saying sustained American English vowels. They measured the fundamental frequency and first three formants (i.e., peaks of the spectral waveform) for each sample and noticed that when plotted in various ways (Figure 2.2), different vowels fell into fairly well-defined different regions of the formant space.

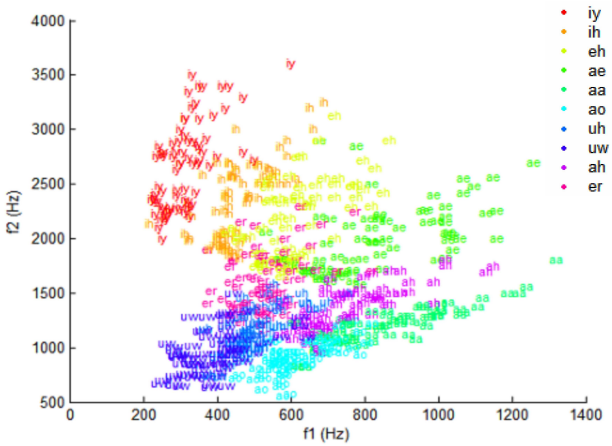


Figure 2.2: The Peterson and Barney 1952 dataset [54]. Displays a scatterplot of the first two formants for the vowels in American English, with different regions labeled by their corresponding vowel categories.

Now, let's consider images of a speaker voicing these same vowels, as shown in Figure 2.3. The face images show the output of a mouth tracking system, where the estimated lip contour is displayed as an ellipse and overlaid on top of the speaker's mouth. The scatterplot in Figure 2.3 shows how a speaker's mouth is represented in this way, with contour data normalized such that a resting mouth configuration (referred to as null in the figure) corresponds with the origin, and other mouth positions are viewed as offsets from this position. For example, when the subject makes an /iy/ sound, the ellipse is elongated along its major axis, as reflected in the scatterplot.

Suppose we now consider the formant and lip contour data simultaneously, as in Figure 2.4. Because the data are conveniently labeled, the clusters within and the correspondences between the two scatterplots are obvious. We notice that the two domains can mutually disambiguate one another. For example, /er/ and /uh/ are difficult to separate acoustically with formants but are easy to distinguish visually. Conversely, /ae/ and /eh/ are visually similar but acoustically distinct. Using these complementary representations, one could imagine combining the auditory and visual information to create a simple speechreading system for vowels.

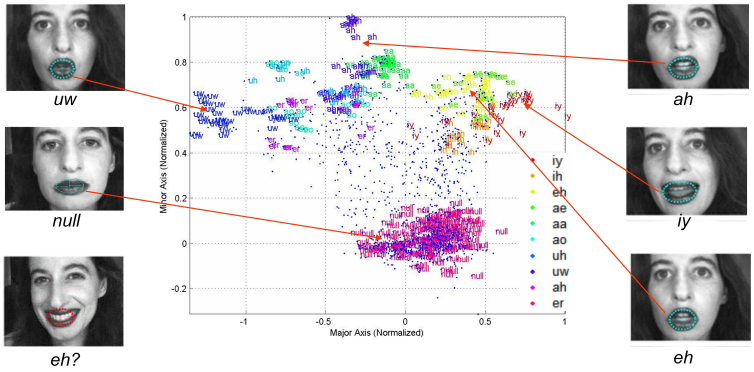


Figure 2.3: Modeling lip contours with an ellipse. The scatterplot shows normalized major (x) and minor (y) axes for ellipses corresponding to the same vowels as those in Figure 2.2. In this space, a closed mouth corresponds to a point labeled null. Other lip contours can be viewed as offsets from the null configuration and are shown here segmented by color. These data points were collected from video of this woman speaking.

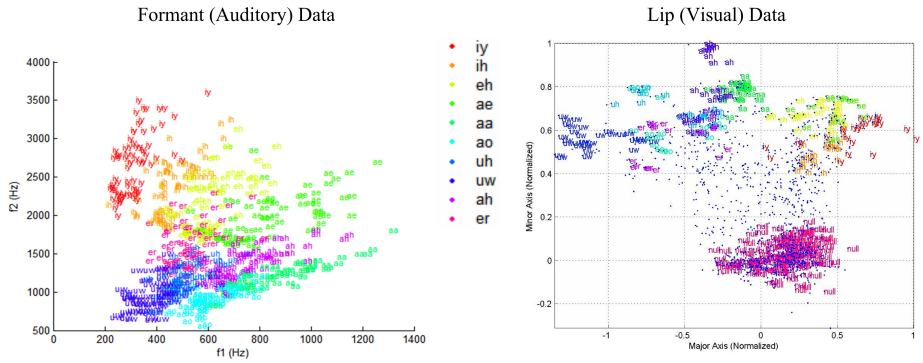


Figure 2.4: Labeled scatterplots side-by-side. Formant data ([54] is displayed on the left and lip contour data (from the author's wife) is show on the right. Each plot contains data corresponding to the ten listed vowels in American English. The region correspondences between the two plots are easy to see.

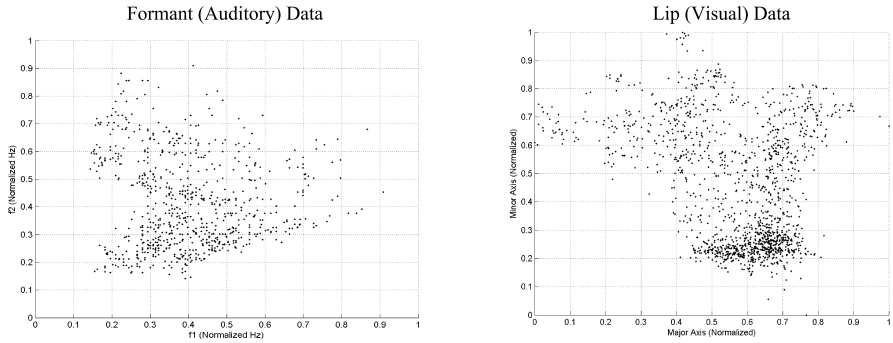


Figure 2.5: Unlabeled data. These are the same data shown above in Figure 2.4, with the labels removed. This picture is closer to what animals actually encounter in Nature. As above, formants are displayed on the left and lip contours are on the right. Our goal is to learn the categories present in these data without supervision, so that we can automatically derive the categories and clusters such as those show above.

2.2.2 Perceptual Grounding: Problem Statement

Given this example, it may be surprising that our interest here is not in building a speechreading system. Rather, we are concerned with a more fundamental problem: *how do sensory systems learn to segment their inputs to begin with?* In the color-coded plots in Figure 2.4, it is easy to see the different represented categories. However, perceptual events in the world are generally not accompanied with explicit category labels. Instead, animals are faced with data like those in Figure 2.5 – which are the exact same data with the labels removed – and must somehow learn to make sense of them. *We want to know how the categories are learned in the first place.*

Why is this difficult?

Solving this type of unsupervised learning problem is difficult because:

1. Nature doesn't label its data. The perceptual inputs animals receive are not generally accompanied by any meta-level data explaining what they represent. Our framework must therefore assume the learning is unsupervised, in that there are generally no data outside of the perceptual inputs themselves available to the learner.

2. Perceptual data are noisy.

3. There is no objective mathematical definition of "coherence" or "similarity."

4. Many perceptual (and motor) systems are inherently dynamic – they involve processes with complex, nonlinear temporal behavior, as can been seen during perceptual bistability, cross-modal influence, habituation, and priming.

2.2.3 Perceptual Interpretation: Problem Statement

The previous section outlined some of the difficulties in unsupervised clustering of nonparametric sensory data. However, even if the data came already labeled and clustered, it may still be challenging to classify new data points using this information. Determining how to assign a new data point to a preexisting cluster (or category) is what we mean by perceptual interpretation. It is the process of deciding what a new input actually represents, and it is classic problem in machine learning.

The problem is exacerbated here because we are interested in biologically realistic representations. We notice how the categories in the above examples display a great deal of overlap. In other words, formant space is very crowded. In classical machine learning, transforming nonseparable samples into higher dimensions is a general heuristic for improving separation with many classification schemes. However, cortical architectures make extensive use of low dimensional spaces, e.g., throughout visual, auditory, and somatosensory processing (e.g., [2]). This was in fact a primary motivating factor in the development of Self Organizing Maps.

Classifying data in these crowded, low-dimensional spaces can be challenging and approaches that try to refine decision boundaries are likely to meet with limited success because there may be no good decision boundaries to find; perhaps in these domains, *decision boundaries are the wrong way to think about the problem.*

Rather than trying to improve classification boundaries directly, one could instead look for a way to move ambiguous inputs into easily classified subsets of their representational spaces. This is the essence of our *influence network* approach. The goal is to use cross-modal information to "move" sensory inputs within their own state spaces to make them easier to classify. Thus, we take the view that perceptual interpretation is inherently a dynamic - rather than static - process that occurs during some window of time. This approach relaxes the requirement that the training data be separable in the traditional machine learning sense; unclassifiable subspaces are not a problem if we can determine how to move out of them by relying on other modalities, which are experiencing the same sensory events from their unique perspectives. This approach is not only biologically plausible, it is also computationally efficient in that it allows us to use lower dimensional representations for modeling sensory and motor data.

2.3 Computational Approaches

In this section, we briefly and non-technically outline our solutions to these three problems:

2.3.1 Perpceptual Grounding

Most of the enormous variability in the world is unimportant. Variations in our sensory perceptions are not only tolerated, they generally pass unnoticed. Of course, some distinctions are of paramount importance and learning which are meaningful as opposed to which can be safely ignored is a fundamental problem of cognitive development. This process is a component of *perceptual grounding*, where a perceiver learns to make sense of its sensory inputs. The perspective taken here is that this is a clustering problem, in that each sense must learn to organize its perceptions into meaningful categories. That animals do this so readily belies its complexity. For example, people learn phonetic structures for languages simply by listening to them; the phonemes are somehow extracted and clustered from auditory inputs even though the listener does not know in advance how many unique phonemes are present in the signal.

Contrast this with a standard mathematical approach to clustering, where some knowledge of the clusters, e.g., how many there are or their distributions, must be known a priori in order to derive them. Assuming that in many circumstances animals cannot know the parameters underlying their perceptual inputs, how then do they learn to organize their sensory perceptions reliably?

To solve this problem, we have introduced the notion of *cross-modal clustering* [7, 8]. This is an approach to clustering based on observed correlations between different sensory modalities. These cross-modal correlations exist because perceptions are created through physical processes governed by natural laws [81, 60]. An event in the world is simultaneously perceived through multiple sensory pathways in a single observer; while each pathway may have a unique perspective on the event, their perspectives tend to be correlated by regularities in the physical world [61]. We propose that these correspondences play a primary role in organizing the sensory channels individually. Based on this hypothesis, we have developed a new framework for grounding artificial perceptual systems. An example of this algorithm learning a mixture of two Gaussian events in the world by way of two complementary sensory modalities is shown in Figure 2.6.

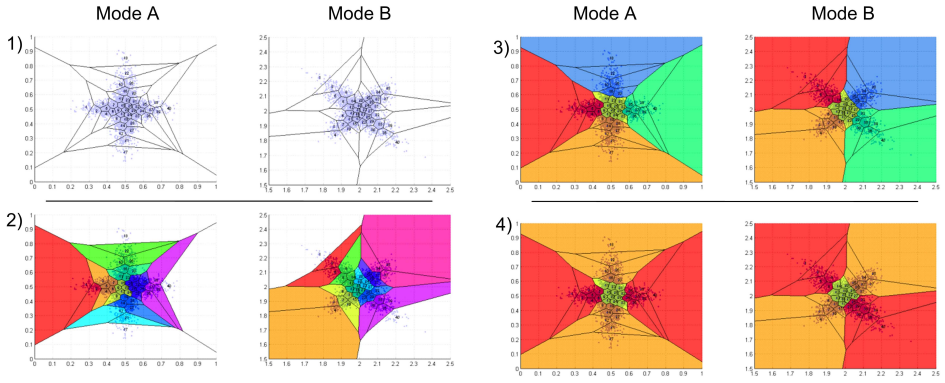


Figure 2.6: The progression of the cross-modal clustering algorithm, learning a mixture of two Gaussians through hypothetical sensory channels *Mode A* and *Mode B*. (1) shows the initial codebook creation in each slice. (2) and (3) show intermediate region formation. (4) shows the correctly clustered outputs, with the confusion region between the categories indicated by the yellow region in the center. Complete details of the algorithm are contained in (Coen 2006a).

The output of the system learning the number (and formant structure) of vowels (monophthongs) in American English is shown in Figure michael:fig:bootstrapping. The system simply watches and listens to someone speaking and then cross-modally clusters the accumulated auditory and visual data. It has no advance knowledge of these vowels and receives no information outside of its sensory channels. This work is the first unsupervised machine

acquisition of phonetic structure of which we are aware.

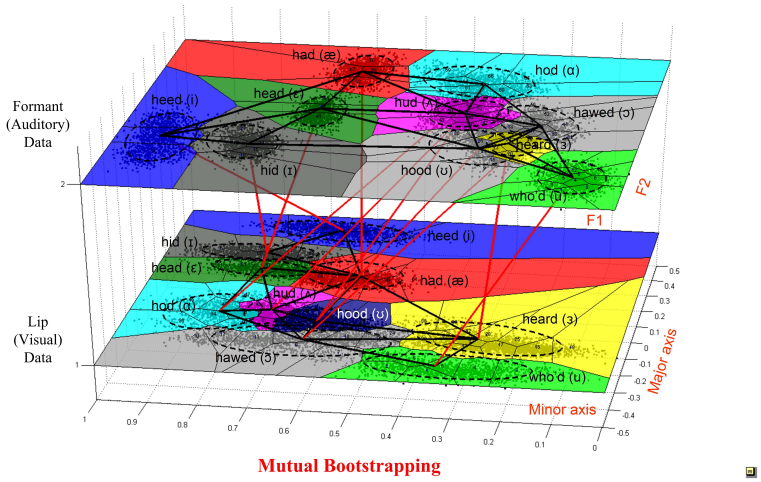


Figure 2.7: Mutual bootstrapping through cross-modal clustering. This displays the formant and lip slices together, where the colors show the region correspondences that are obtained from cross-modal clustering. Initially, the modes "knew" nothing about the events they perceive. Cross-modal clustering lets them mutually structure their perceptual representations and thereby learn the event categories that generated their sensory inputs. The black lines in the figure connect neighboring regions within each mode and the red lines connect corresponding regions between them. The identifying labels were manually added for reference and ellipses were fit onto the regions to aid visualization. All data have been normalized.

2.3.2 Perceptual Interpretation

In the previous section, we saw that sensory systems can mutually structure one another by exploiting their spatiotemporal co-occurrences. We called this process of discovering shared sensory categories *perceptual grounding* and suggested that it is a fundamental component of cognitive development in animals; it answers the first question that any natural (or artificial) creature faces: what different events in the world can I detect?

The subject of this section follows naturally from this question. Once an animal (or a machine) has learned the set of events it can detect in the world, *how does it know what it is perceiving at any given moment?* We refer to this as perceptual interpretation. We will take the view that perceptual interpretation is inherently a dynamic - rather than static - process that occurs during some window of time. This approach relaxes the requirement that our perceptual categories be separable in the traditional machine learning sense; unclassifiable subspaces are not a problem if we can determine how to move out of them by relying on other modalities. We will argue that this approach is not only biologically plausible, it is also computationally efficient in that it allows us to use lower dimensional representations for modeling sensory and motor data.

We have formulated a dynamic model of perceptual interpretation called *influence networks* that allow co-occurring modalities to influence each other while they are in the midst of perceiving. This network is illustrated in Figure 2.8, where two independent perceptual channels are interconnected such that they can mutually reinforce one another. The dynamics of this network are described via a set of state equations on a leaky integrate and fire network as described in [8]. The activity of this network is illustrated a simple speechreading system in Figure 2.9.

2.3.3 Sensorimotor Learning

Up to this point, we have been concerned with learning to recognize events in the world. We now turn to the complementary problem of learning to generate events in the world. That these two problems are interrelated is well established - animal behaviors are frequently learned through observation, particularly in vertebrates.

Our approach is to recursively reapply the perceptual framework presented above. We will treat the motor

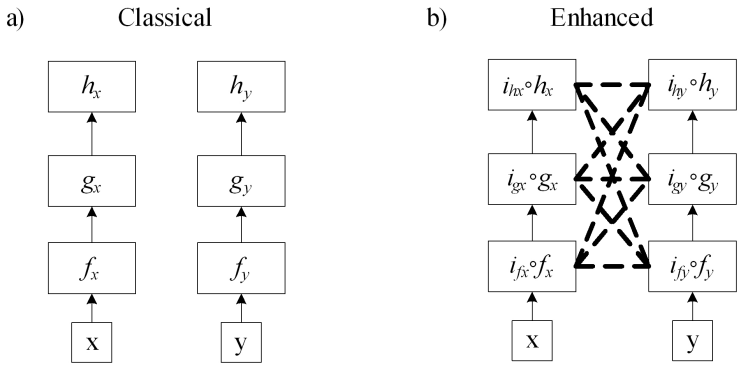


Figure 2.8: Adding an *influence network* to two preexisting, independent systems. We start in (a) with two pipelined networks that independently compute separate functions. In (b), we compose on each function a corresponding influence function, which dynamically modifies its output based on activity at the other influence functions. The interaction among these influence functions is described by an influence network, which is defined in (Coen 2006a). The parameters describing this network can be found via unsupervised learning for a large class of perceptual systems, due to correspondences in the physical events that generate the signals they perceive and to the evolutionary incorporation of these regularities into the biological sensory systems that these computational systems model. Note influence networks are distinct from an unrelated formalism called influence diagrams.

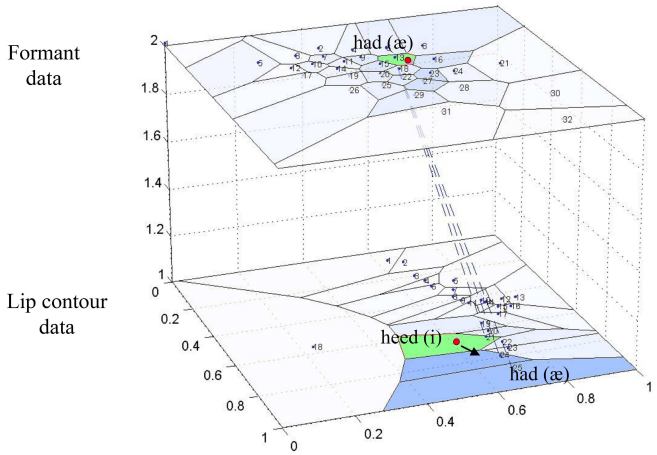


Figure 2.9: Cross-modal activations in an influence network. The auditory perception in formant space on top increases activations in the lip contour space on the bottom. This example is unusual because it shows an auditory modality influencing a visual one, which is biologically realistic but unusual in an artificial perceptual system. The effect of this influence is that the visual perception is modified due to an induced Hebbian gradient. Shading here corresponds to activation potential levels.

component of sensorimotor learning as if it were a perceptual problem. This is surprising because one might suppose that motor activity is fundamentally different than perception. However, we take the perspective that motor control can be seen as perception backwards. We imagine that - in a notion reminiscent of a Cartesian theater - an animal can ”watch” the activity in its own motor cortex, as if it were a privileged form of internal perception. Then for any motor act, there are two associated perceptions - the internal one describing the generation of the act and the

external one describing the self-observation of the act, as shown in Figure 2.10. The perceptual grounding framework described above can then cross-modally ground these internal and external perceptions with respect to one another, as in Figure 2.11. The insight behind this approach is that a system can develop motor control by learning to generate the events it has previously acquired through perceptual grounding, as in Figure 2.12.

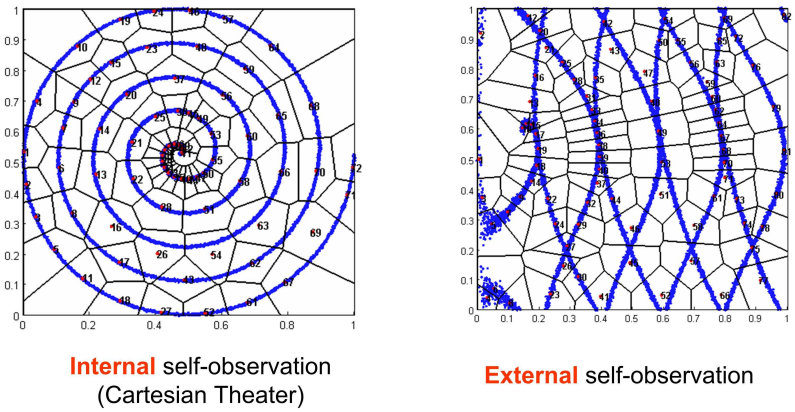


Figure 2.10: Two views of motor behavior. On the left, we have internal perception of exploratory motor behavior corresponding to an Archimedean spiral. These data correspond to the parameters used to generate nascent motor activity. On the right, we see the corresponding external perceptions of these exploratory motor behavior. These data correspond to perceptual features describing sensory observations.

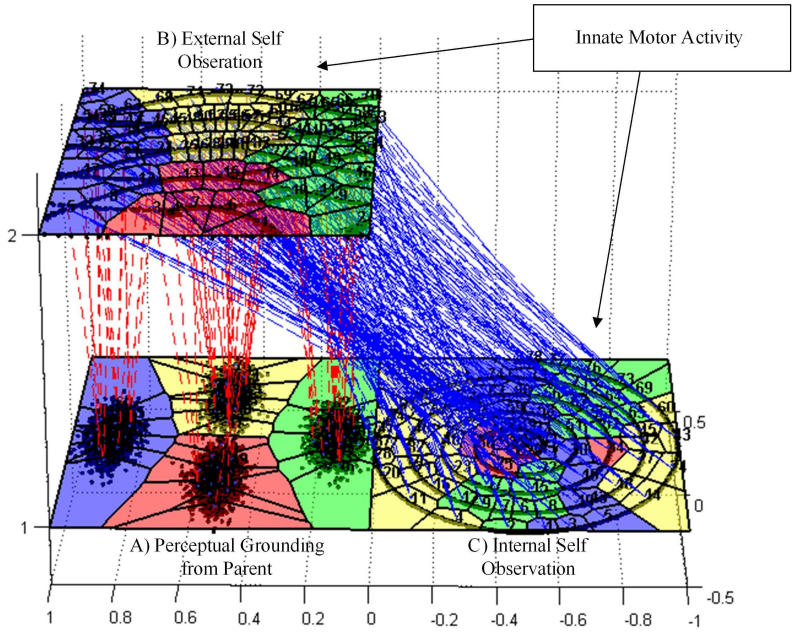


Figure 2.11: Stages of recursive cross-modal clustering. Starting from the acquisition of perceivable events in (A), we learn to classify the effects of our own behaviors in terms of these events in (B). Finally, we can then relate this back to the innate motor activity generating our actions, as in (C). There are thereby three stages of cross-modal clustering in this model.

A benefit of this framework is that it can learn imitation, a fundamental form of biological behavioral learning

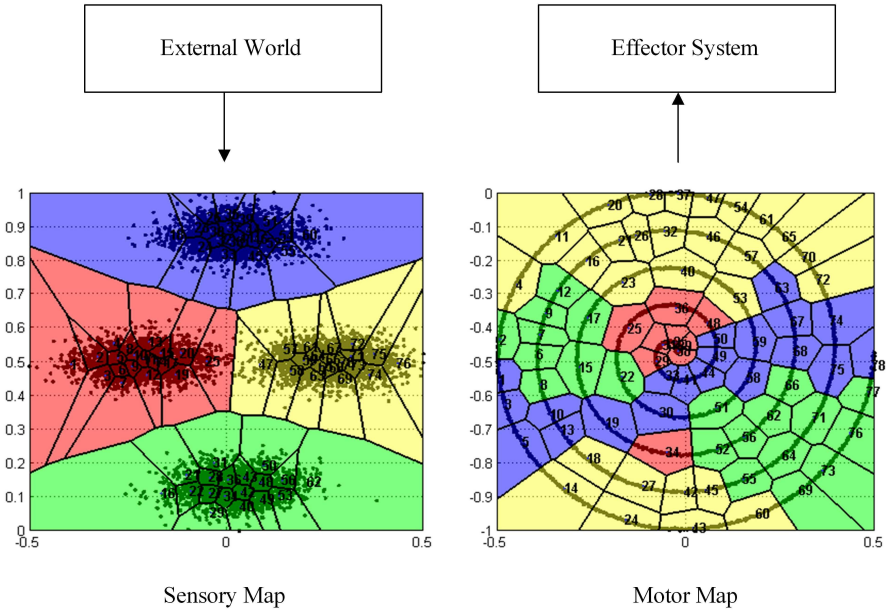


Figure 2.12: Acquisition of voluntary motor control. Regions in the motor map on the right are now labeled with the perceptual events they generate in the sensory map on the right. This is achieved via the process outlined in Figure 2.11.

(Meltzoff and Prinz 2002). In imitative behaviors - sometimes known as mimicry - an animal acquires the ability to reproduce some aspect of another's activity, constrained by the capabilities and dynamics of its own sensory and motor systems. This is widespread in the animal kingdom [3]. and is thought to be among the primary enablers for creating self-supervised intelligent machines [11].

We demonstrate sensorimotor learning in this framework with an artificial system that learns to sing like a zebra finch. Our system first listens to the song of an adult finch; it cross-modal clusters this input to learn songemes, primitive units of bird song that we propose as an avian equivalent of phonemes. It then uses a vocalization synthesizer to generate its own nascent birdsong, guided by random exploratory motor behavior. The motor parameters describing this exploratory vocal behavior are treated as if they correspond to external perceptual inputs. By simultaneously listening to itself sing, the system organizes its motor maps by cross-modally clustering them with respect to the previously learned songemes of its parent. During this process, the fact that the motor data were derived internally from innate exploratory behaviors, rather than from external perceptual events, is irrelevant. By treating the motor data as if they were derived perceptually, the system thereby learns to reproduce the same sounds to which it was previously exposed. This approach is modeled on the dynamics of how male juvenile finches learn birdsong from their fathers [80, 21].

These developmental learning steps are outlined in Figure 2.13. The resulting birdsong mimicry learned via this system illustrated in Figure 2.14.

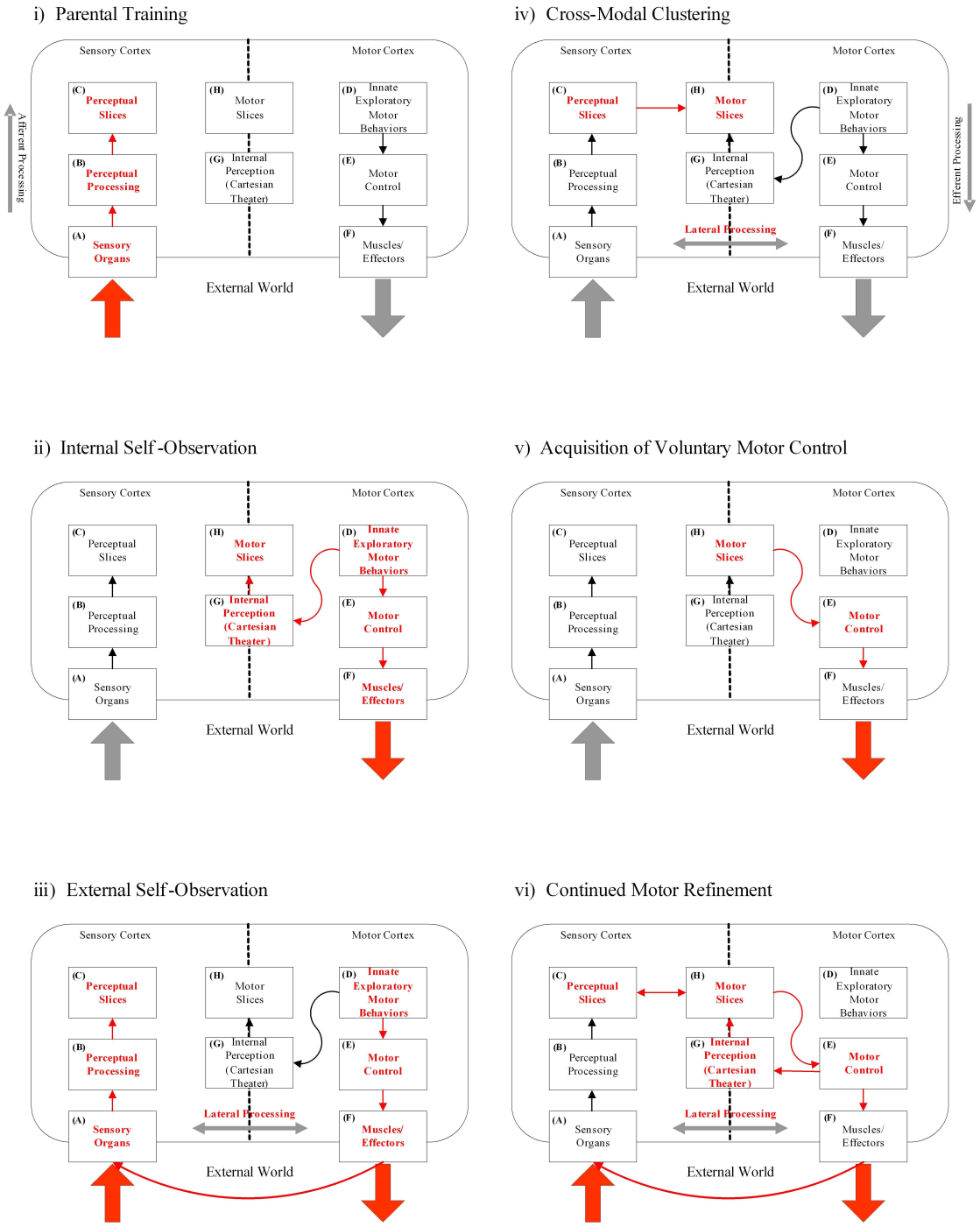


Figure 2.13: Developmental stages in our model. i) The juvenile acquires perceptual structures from its parent. ii) Motor acts are observed internally through a Cartesian Theater. iii) The effects of motor acts are observed externally through perceptual channels. iv) Motor slices are cross-modally clustered with respect to perceptual slices. The juvenile thereby learns how to generate the events it learned in stage (i). v) Random exploratory behaviors are disconnected and motor slices take over the generation of motor activity. The juvenile is now able to intentionally generate the sensory events acquired from its parent. vi) Internal perception can be used subsequently in non-juveniles to refine motor control.

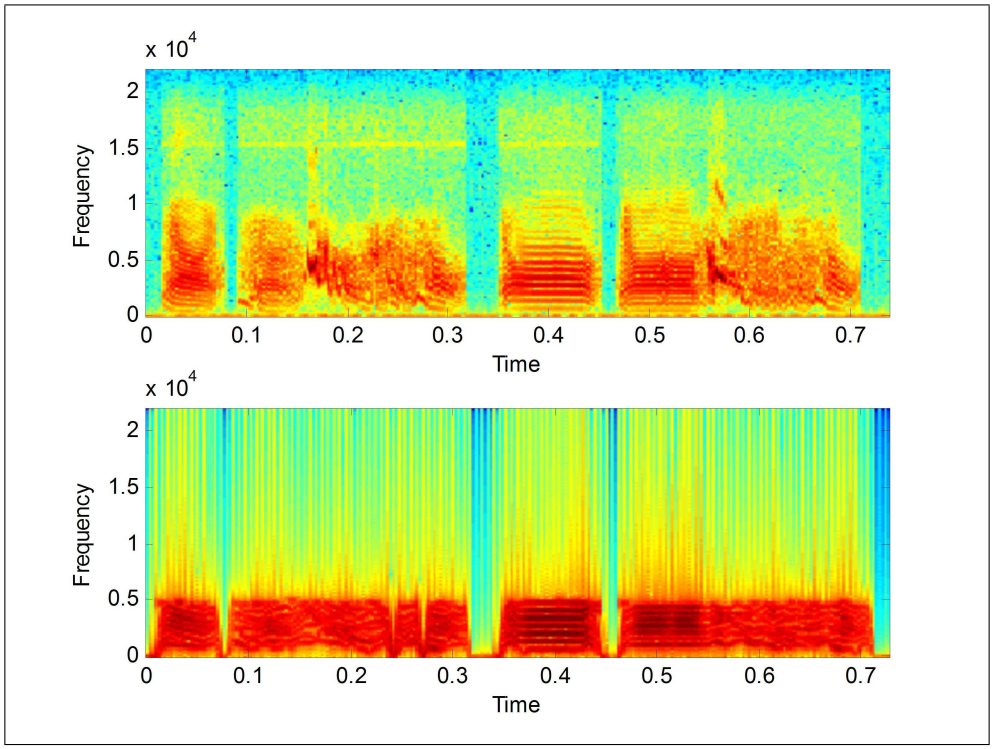


Figure 2.14: Birdsong mimicry. On the top is a sample of the real zebra finch song used as the ”parent” for our system. On the bottom is the system's learned imitation, where the acquired songemes have been fit to the template of the parent's song and smoothed.

2.4 Learning the contents of episodic, semantic, and plan memories

At the most abstract level, psychological research has distinguished between two main memory systems: episodic and semantic memory. CHIP also distinguishes a third system, plan memory, which is responsible for storing abstract "how-to" knowledge. Each of these is responsible for representing knowledge applicable to different aspects of experience. Episodic memory divides knowledge into temporally coherent chunks, providing the contextual cues that support recognition and understanding of the world. Semantic memory stores knowledge about the meaning of the world: the roles of and relations between concepts. Semantic knowledge supports inference by specifying the causal roles that entities may play, thereby outlining the set of possible states that may occur, given knowledge of the current state. Plan memory caches knowledge about how to act on the world, supporting recognition of others actions, and facilitating interventions that change the state of the world.

Two representational principles guide our approach to modeling human-level learning across the episodic, semantic, and planning systems: *relational learning* and *clustering*. Applied to multi-modal experience, the principle of relational learning suggests that models should be capable of learning relations across different modes, when relevant. Applied to temporally structured experience, the principle of clustering suggests that models should be able to take a continuous stream of data and cluster temporally contiguous states into episodes, where episodes contain experiences of a similar kind. Applied to abstract concepts, the principles of relational learning and clustering suggest that models should be able to discover abstract concepts and their relations. Applied to sequences of actions, these principles suggest that models should learn to cluster actions, and learn relations between simple actions to form more complex planning units.

These principles are implemented within a probabilistic, hierarchical, generative framework, which we refer to as the *theory-based Bayesian framework*. Theory-based Bayesian models of induction focus on three critical questions: what is the content of probabilistic theories, how are they used to support rapid learning, and how can

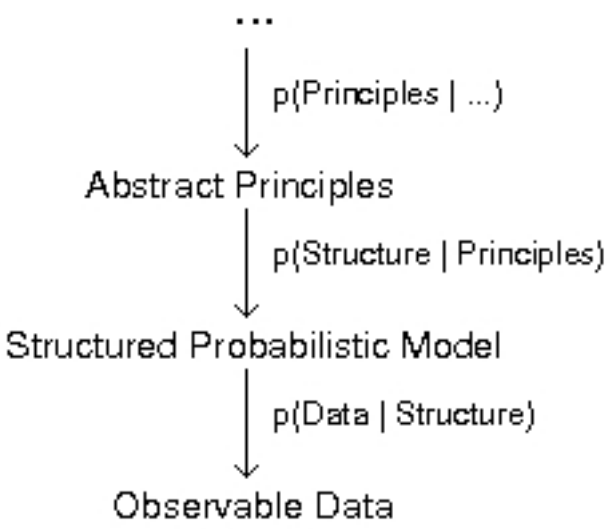


Figure 2.15: The theory-based Bayesian framework that forms the basis of CHIP's high-level memory architecture. Observable data is generated by structured probabilistic models, such as clusters, relations between clusters, graphs (trees, lines, etc./), or a particular grammar. Structured models are, in turn, generated by abstract principles. Clusters and relations are generated by abstract principles such as the preference for relatively small numbers of categories and relations, graphs are generated by principles of structural form encoded in graph growing grammars, and grammars are generated by grammar grammars, which provide abstract templates for the rules of generative grammars. Given data, inferences can be made up the hierarchy, for example, at the structure level inferring the probability that a particular tree structure generated the data, and at the principle level, the probability that tree is the best structural form.

they themselves be learned? The learner evaluates hypotheses h about some aspect of the world – the extension of a property or category, or the presence of a hidden cause – given observed data x and subject to the constraints of a background theory T. Hypotheses are scored by computing posterior probabilities via Bayes' rule:

$$P(h|x,T) = \frac{P(x|h,T)P(h|T)}{\sum_{H' \in \mathcal{H}_T} P(x|h',T)P(h'|T)}. \tag{2.1}$$

The likelihood $P(x|h,T)$ measures how well each hypothesis predicts the data, while the prior probability $P(h|T)$ expresses the plausibility of the hypothesis given the learner's background knowledge. Posterior probabilities $P(h|x,T)$ are proportional to the product of these two terms, representing the learner's degree of belief in each hypothesis given both the constraints of the background theory T and the observed data x. Adopting this Bayesian framework is just the starting point for our memory architecture. The challenge comes in specifying hypothesis spaces and probability distributions that support Bayesian inference for a given task and domain. In theory-based Bayesian models, the domain theory plays this critical role.

More formally, the domain theory T generates a space $\mathcal{H}_T$ of candidate hypotheses, such as all possible extensions of a property, along with the priors $P(h|T)$ and likelihoods $P(x|h,T)$. Prior probabilities and likelihoods are thus not simply statistical records of the learner's previous observations. Neither are they assumed to share a single universal structure across all domains. Rather, they are products of abstract systems of knowledge that go substantially beyond the learner's direct experience of the world, and can take qualitatively different forms in different domains.

We will distinguish at least two different levels of knowledge in a theory (Figure 2.15). While fully articulated domain theories may be much richer than this picture suggests, we focus on the minimal aspects of theories

needed to support inductive generalization to demonstrate our approach. The base level of a theory is a structured probabilistic model that defines a probability distribution over possible observables – entities, properties, variables, events, or actions. This model is typically built on some kind of structure capturing relations between observables, such sets of mutually exclusive categories, a taxonomic hierarchy or a causal network, together with a set of numerical parameters. The graph structure determines qualitative aspects of the probabilistic model; the numerical parameters determine more fine-grained quantitative details. At a higher level of knowledge are abstract principles that generate the class of structured models a learner may consider, such as the specification that a given domain is organized taxonomically or causally. Inference at all levels of this theory hierarchy (Figure 2.15) – using theories to infer unobserved aspects of the data, learning structured models given the abstract domain principles of a theory, and learning the abstract domain principles themselves – can be carried out in a unified and tractable way with hierarchical Bayesian models [25].

2.4.1 Episodic memory: segmenting experience

Episodic memory captures information in the temporal structure of experience; characteristic scenes and scenarios that co-occur in the world. From the perspective of artificial intelligence, exploiting episodic structure is crucial for dealing with the large knowledge bases associated with real-world reasoning problems. Given the wealth of knowledge people possess, it seems clear (1) that people can't consider all of this at once, (2) it doesn't matter because only a small subset of knowledge is relevant at any given time. Episodic structure provides an opportunity to guide knowledge retrieval: if in a kitchen, one is likely to need knowledge about stoves, fire, spatulas, and the location of spices, but unlikely to need knowledge about carborators or bicycles. Indeed, considerable research involving on memory suggests that people's knowledge is stored and retrieved in part based on context [e.g. 31].

CHIP's episodic memory system implements the principle of clustering to identify segments of experience that have the same content. A simple version of this mechanism has been applied to sequential data. The mechanism looks for segments that contain similar kinds of content. Importantly, this approach simultaneously learns the content of the sequence (topics), while finding episodes that are consistent with respect to these inferred topics, thus resolving a potential chicken-or-egg problem by learning semantic and episodic structure together. Consider for example a typical meeting, with people discussing an agenda containing several different issues. From a transcription, the structure of the meeting is not immediately evident because the issues run together; but by reading the transcript, people are quite able to parse it into segments that correspond to different issues. Figure 2.16 shows the best set of topics and segmentations found by the model (and by people) for an example meeting. Given these different episodes, one can predict that certain topics and therefore words will be more or less likely, and exploiting this context, polysemous words may be disambiguated.

This demonstration provides insight into how the general principle of clustering may be implemented in a mechanism that segments sequences of words into episodes. Similarly, this mechanism can be applied to sequences of visual scenes for which visual features or objects have been identified, eliciting context effects in recognizing objects. Additionally, it may be applied to concurrent streams of scenes and text, exploiting the co-occurance of words with visual context.

This mechanism is one implementation of learning by generalization in a hierarchical generative framework. Experiences are generated by a simple model that assumes that an episode is characterized by similar individual experiences, which are grouped by topic. Sequences are generated by choosing a set of (distribution over) topics for the current episode. Then, for each individual experience, a topic is selected and a particular word is chosen from that topic. Given a continuous string of experience, the model then searches for segmentations that that allow the simplest way of summarizing the semantic structure.

However, this mechanism has only used the principle of clustering to group scenes that are locally consistent into episodes. Clearly, learning relations between adjacent scenes is important as well, otherwise in would be difficult to predict how experience unfolds in time. In the next two sections, we address how to learn causal/relational structure between abstract concepts, which provides constraints on how the world can change and how to learn abstract plans, which facilitate acting on the world as well as providing scaffolding for understanding events in terms of

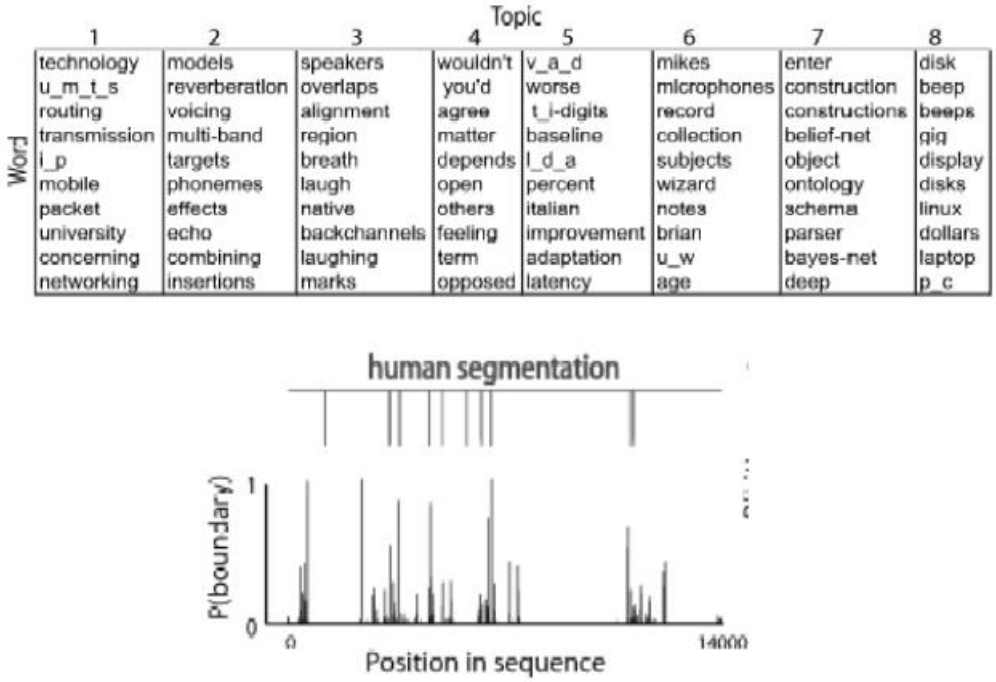


Figure 2.16: The best topics and segmentations found for a meeting from the ICSI database. The ten most indicative words for each topic are shown on the top. On the bottom is a comparison of episodes predicted by the model and humans.

other's actions. Together, these will guide understanding of how experience unfolds in time, and provides feedback regarding how to parse experience.

2.4.2 Semantic memory: roles of and relations between concepts

Semantic memory captures information about regularities that underlie the structure of the world: concepts, kinds, and lawful relations among them. Much research in psychology has sought to understand how concepts at varying levels of abstraction such as pigeon, bird, animal, and biology are learned, how concepts within each of these levels are related, and how different levels of abstraction are related. Though many of the details remain unresolved, several main themes have emerged. First, representations are flexible, with multiple ways of organizing the entities, each explaining different aspects of the entities in the domain. Second, different domains are characterized by different structural forms which describe how entities are related. Third, entities may be grouped into concepts which play different roles, standing in characteristic relationships with each other.

These competencies represent the building blocks for a semantic memory system that identifies objects as belonging to (potentially many) kinds, where kinds are characterized by the kinds of relations that they may enter into. The goal of this system is then to constrain predictions about how the world may change, given knowledge of the current world state. We discuss models that demonstrate the four basic competencies by exploiting the principles of relational learning and clustering in a generative hierarchical framework.

Learning flexible representations

People can think about entities in many ways; for example, a bear is both a mammal and a land predator. Different ways of thinking about an entity support different ways of reasoning about the world: if one learned about a novel gene that was found in bears, one might reasonably infer that other mammals, such as whales would have the gene. However, if one learned about a novel behavior, for example, that bears tend to circle before sleeping, it seems less likely that whales would have the same property because it seems more relevant to other land animals. People's ability to flexibly reason is a key aspect of human intelligence, and underlies the kinds of relevance-based inference that support everyday cognition [76, 43].

CHIP's semantic memory system incorporates the ability to flexibly learn multiple ways of categorizing entities. It assumes that in a domain such as animals, there are different kinds of features each of which implies a particular way of categorizing the entities in the domain. An important characteristic of this system is that the model is able to learn how many different kinds of features there are, how many categories to create for each kind, as well as assignments of features to kinds and objects to categories. Demonstrating these capabilities, we have implemented such a model on a binary matrix encoding whether 22 animals have 106 features. The best solution is displayed in Figure 2.17. The model finds three kinds of features: taxonomic, noisy/irrelevant, and ecological. The categories discovered are appropriate: animals are categorized into taxonomic categories such as mammals, birds, reptiles/amphibians, and invertebrates and ecological categories such as land predators, aerial predators, aquatic predators and prey. The model also provides insight into how people may determine the relevance of different kinds of knowledge in a given situation. If informed about a novel property that was true of bats and dragonflies, the model predicts that ecological knowledge is relevant to predicting whether the feature is more likely to be true of owls or ostriches. Interestingly, the model also "imagines" things it has not seen before; for example, it knows what properties a flying reptile would have, even though it has seen no such thing.

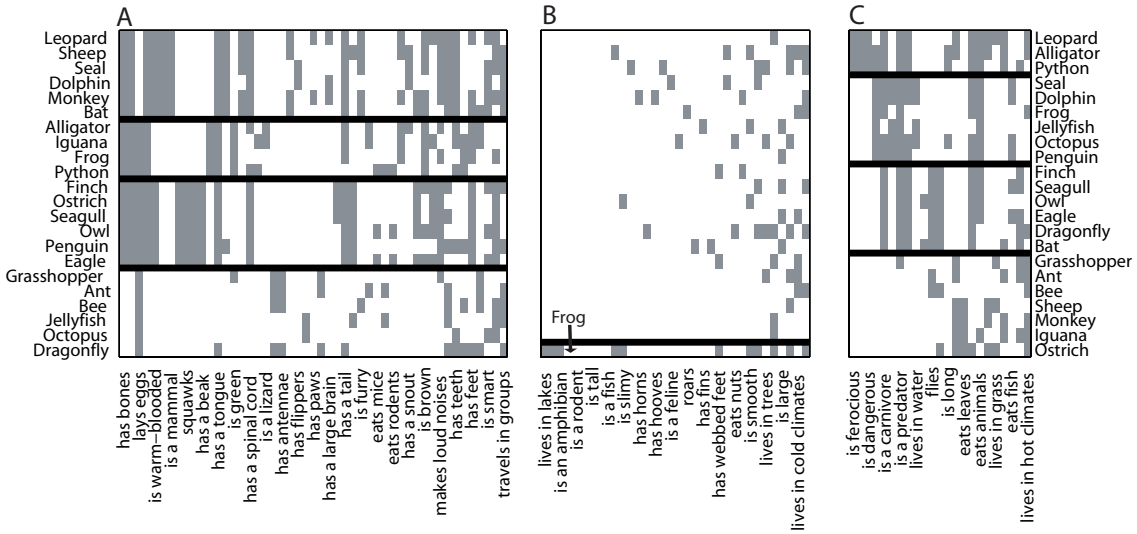


Figure 2.17: The best solution found for the animals data includes three systems of categories: (a) a taxonomic system, including categories of mammals, birds, reptiles/amphibians, and invertebrates (b) a set of uninformative features and (c) an ecological system, including categories of land predators, aerial predators, marine predators, and prey. Objects are labeled for the taxonomic and ecological systems. Features are presented in order of decreasing predictability. Due to space constraints, only every other feature is labeled.

The demonstration leverages the principle of clustering into learning multiple domain representations, capturing the basic intuitions that categories may be both mutually exclusive (e.g. mammals and fish) or overlapping (e.g. fish and pets). By allowing multiple representations, it supports flexible relevance-based reasoning. By incorporating the assumption that each feature belongs to a single cluster, it can imagine things it has never seen before, and exploits

informational encapsulation to allow modular updating of different kinds of knowledge.

Learning structural form

Different domains may have different structural forms which characterize relationships among entities: political tendencies are characterized by positions on the left or right of a line, taxonomic relations among animals are characterized by a tree, geographic relations are characterized by a grid, and days and seasons are characterized by cycles. The ability to learn the abstract form that underlies a domain is important because it provides an important constraint on how we think about the world. If introduced to a novel animal, one knows it may be subordinate to mammals or birds but not both because this would violate the tree constraint. Similarly if introduced to a new set of animals, one may reason by analogy that the relations among these animals may also be characterized by a tree structure.

The CHIP architecture incorporates the ability to learn that different structural forms apply to different domains, abstract knowledge that can be used to support analogical reasoning about novel domains. In this implementation, we utilize a hierarchical framework which contains at the most abstract level different graph grammars. Different grammars give rise to different graph structures, such as trees, and lines, and rings at the next level down. Finally, we can imagine data being generated over a particular structure by a mechanism that randomly generates features which are smooth with respect to the graph. Then, given knowledge about the features of entities in the domain, we can infer the graph that was most likely to generate the observed distribution of features. To demonstrate these capabilities, the model was run in two domains, animals and voting records of judges. The results are shown in Figure 2.18, where we see that the model learns that a tree structure applies to the domain of animals, and a line representing conservative and liberal voting records characterizes justices. The model has learned that if a new justice comes along, they will either tend to vote with Marshall or Powell, but not both, and if a new case comes along, justices Marshall and Brennan are more likely to vote together than Marshall and Stevens. Importantly, the model has also learned the abstract principle that the domain of politics is governed by a line: if introduced to an entirely new set of judges, the model could analogically predict that a line structure is more likely to fit their voting patterns than a tree. Importantly, this analogical reasoning mechanism is not constrained to have a precise mapping between the two domains: the novel domain may not have a precise one-to-one mapping, as long as the same generative principles can be applied.

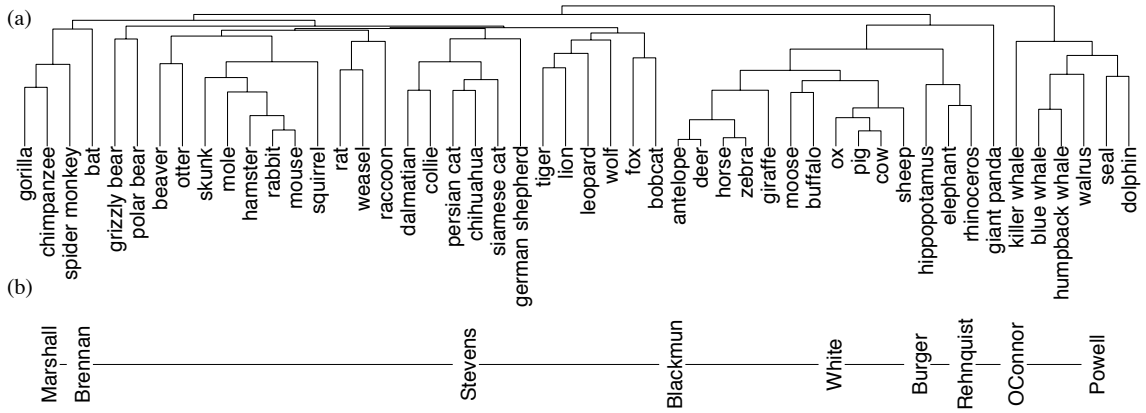


Figure 2.18: The best solution found for the domains of (a) animals and (b) judicial decisions. For animals, a tree structure including subtrees corresponding to primates, bears, rodents, canines and felines, ungulates and marine mammals best describes the domain. For judicial decisions, a line representing the spectrum from liberal to conservative justices best describes the domain.

This model also demonstrates how the principles of relational learning and clustering can be used to implement richly structured knowledge. Relationships among objects in a domain may be characterized by structural relations which cluster objects together in different ways. Tree relations form hierarchically related sets of clusters over

objects, while linear relations form overlapping clusters without hierarchical relational structure.

Learning causal/relational schemas over kinds

People learn causal relational structures over abstract kinds: we know that coal mining is a behavior, lung cancer is a disease and that coughing blood is a symptom. We also know that behaviors cause diseases, which cause symptoms. This is an important discovery because it constrains what people will posit to be true in the world. If one learned about a novel behavior and symptom that tended to co-occur, one should not posit a direct link between a behavior and a symptom, but rather infer that the behavior causes a disease (known or unknown) which in turn causes a symptom, even though hypothesizing a hidden disease is more complex than hypothesizing direct link from the symptom to the disease.

CHIP's hierarchical, generative framework can be used to learn these abstract conceptual relations. Imagine a clustering of the entities into concepts, and a set of arrows indicating relations between concepts, where an arrow indicates a relation tends to hold between the concepts. From data representing the different relations that hold between different entities, relations and entities can both be clustered, discovering kinds of entities that enter into the same kinds of relations, as well as kinds of relations that are characterized by applying in the same ways to the same sets of entities. To demonstrate the efficacy of the model, it was run on a medical dataset including entities such as algae, carbohydrate, enzyme and disease and relations such as affects, interacts with and causes. Figure 2.19 shows the best solution. The model has learned, for example, that biological functions affect organisms and chemicals cause diseases. As in previous examples, this abstract knowledge constrains analogical inferences about novel objects and novel relations, and transfers to novel sets of objects and relations and relations from the same domain.

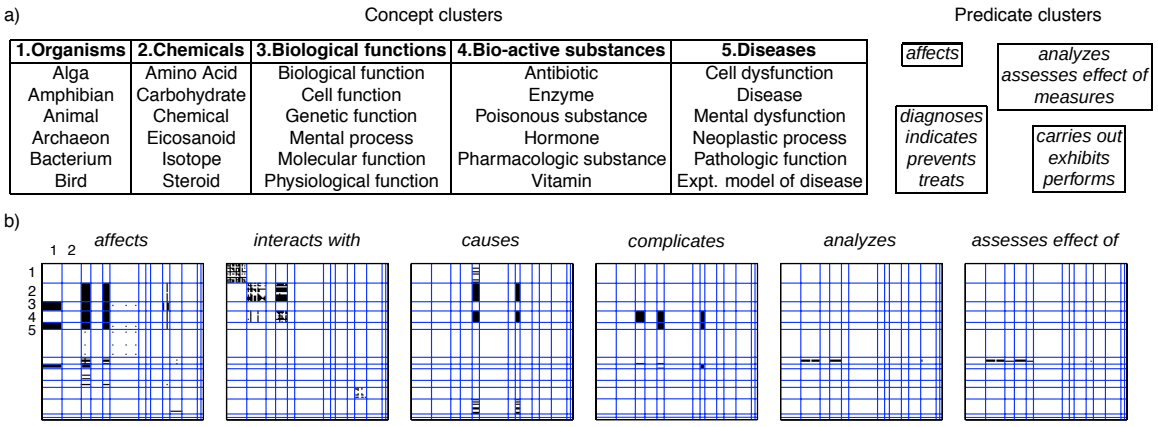


Figure 2.19: Learned relational structure for UML ontology. The best solution shows clusters of concepts including organisms, chemicals, and biological functions and predicate clusters of corresponding to relations such as affects and measurement. The model has learned, for example, that biological functions affect organisms and that chemicals cause diseases.

This model shows how the principles of clustering and relational learning can be used to learn abstract concepts and ways in which these concepts may interact. Importantly, the abstract concepts that are discovered are only defined in terms of the ways in which they relate, but have implications for understanding how to intervene on the world.

Semantic knowledge: summary

Learning semantic knowledge is a problem of learning abstract kinds and relations that are similar with respect to changes that may happen in the world. In this section, we have focused on unsupervised learning of this abstract knowledge, but learning by instruction may be implemented as a special case where knowledge structures are directly modified, given input from a teacher. Our demonstrations have moved from the relatively concrete issue of learning flexible representations for single domains to relatively abstract knowledge about how to discover kinds and their

relations. Each demonstration has used the principles of clustering and learning relations, and in the process we have seen that these basic principles support extensions to richly structured representations and multiple levels of abstraction. Additionally, the hierarchical generative framework provides natural support for analogical reasoning in a setting where analogies are constrained by abstract generative principles rather than exact one-to-one mappings. These models fall far short of a full description of semantic knowledge, addressing only some fundamental issues, but demonstrate the promise these simple principles for modeling the range of semantic knowledge.

2.4.3 Plan memory: learning how to change the world

Plan memory contains action representations, encoded routines for intervening on the world. We take the role of plan knowledge to be be somewhat abstract [44, 63], assuming discrete, basic actions such as walking, turning right, and picking up a can, and more complex compositions such as going to the store and shopping. Because plan knowledge is the basis for intervening on the world, it is critical for any active agent. It also provides an opportunity to provide feedback to semantic knowledge: given the current state of the world, semantic knowledge dictates what next possible states are achievable. With knowledge of how to affect change, an agent may test hypotheses about how to change the world and update semantic knowledge accordingly.

Research in psychology has characterized action representations as hierarchical, composed of abstract units which are in turn composed of basic actions [13, 36]. We have recently begun investigating probabilistic generative grammars for modeling this kind of hierarchically nested knowledge structures in the CHIP architecture. For example, consider the case of the abstract action "going shopping" which may be cashed out in three parts, "going to the store", "shopping", and "returning home". Eventually, these could terminate in simple action units such as "walking straight", "starting the car" etc. The model, given discrete sequences of actions, would able to learn abstract action clusters, like "shopping" as being composed of "finding soup", "put in the cart", etc. as well as which actions and action clusters are likely to follow others.

These generative representations could then be used to infer plans from experience, using Bayesian inference. Given sequences of actions, the plan memory system would extract hierarchical regularities or chunks, observing that walking a particular route is associated with the higher order plan unit of "going shopping" while another route would be associated with the "going to the subway". Importantly, the system would be able to recognize if subparts of these plans overlapped and reuse these units allowing simpler representations by composing plans out of commonly used subplan chunks. In addition, these action units could then be used to support recognition of others actions and parsing of events and episodes [1, 49, 24]. This work is currently in progress and results for large scale simulations are planned for future work.

2.5 Conclusions

The CHIP architecture provides a framework for an integrated system for learning. At the level of parameter memory, CHIP employs cross-modal clustering to learn how to distinguish between the basic sensory constituents. It uses influence diagrams, bi-diretional pipelines with cross modal clustering to understand perceptions, even in the spite of noise. Final, it uses its ability to perceive its internal state in order to train its motor system to imitate events it sees in the world.

At the higher level, CHIP provides a framework for learning from sequences of multi-modal data. We have identified three areas, inspired by psychological research, which frame CHIP's abstract memory architecture: episodic, semantic, and plan memory. In this hierarchical, generative architecture, episodic, semantic, and plan regularities are learned from experience. Representations of these regularities are stored in memory and subsequently used to constrain inferences in the world. Episodic knowledge constrains what semantic and plan is needed at any time. Semantic knowledge constrains and guides reasoning about what is possible, given the current state and what actions are potentially useful now. Plan knowledge facilitates intervention on the world, and recognition of other's actions.

CHIP's high level representations are based on two simple principles, relational learning and clustering, implemented within a hierarchical probabilistic framework that form the basic structure of our memory systems across these domains. We have demonstrated how these principles can be leveraged to learn episodic structure in text, rich representations of domain knowledge, causal regularities between abstract kinds, and proposed extensions to

modeling action representation. The demonstrations we have presented do not cover the breadth of functions human memory serves, but guided by our basic principles we believe that we will continue to make progress toward an architecture that more closely matches the capabilities of human learners.

Chapter 3

Perception

What role does perception play in human cognition? and How might we engineer a system with similar capabilities?

Evidently, sensorimotor processes help us perceive the world and act on it, and have evolved to deal with the "here-and-now" aspect of mundane behavior. Given that humans share highly sophisticated sensorimotor capabilities with animals one could easily conclude that perception and action are mere I/O channels, and the magic of human cognition must lie somewhere else. Previous approaches to engineering human-like cognition have taken exactly this position. In the sense-model-plan-act model of cognition adopted by shown on the left in Figure 3.1 perceptual systems are simply a front end that provide symbolic descriptions of the world that feed into planning or reasoning systems which, in some cases pass on their decisions to a motor controller. The architecture we have adopted (shown on the right in Figure 3.1) takes a very different approach and highlights aspects of perception that are far more consistent with what we have learned from biology, namely:

1. There are qualitatively 3 different kinds of perceptual representations and processes involved in cognition. At the lowest level *body-based sensorimotor* representations are devoted to modeling the body in the world. Examples include; topographic maps of the body and the consequences of one's own actions on one's own sensors (e.g the fact that moving ones arm will cause predictable changes in one's proprioceptive state, and cause the arm to appear at precise locations in the visual field). These representations are crucial for simply being in the physical world and to distinguish ourselves from it. They start getting populated immediately after birth and are always active so as to be adaptive to growth and injury. Sophisticated learning mechanisms are responsible for discovering the

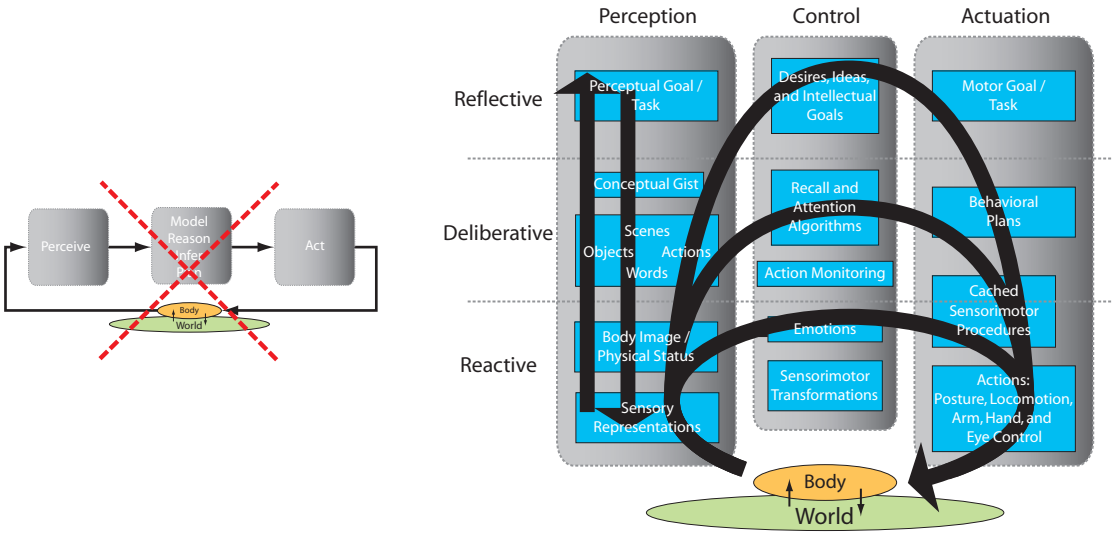


Figure 3.1: Unlike traditional production-rule systems (on the left) where perceptual processes are a front-end to symbolic reasoning, they are involved at every level of our architecture (on the right) through layered loops that tightly couple perception, decision making and action

cross-modal correspondences. See 2 for one such algorithm.

At the next level up, there exist a set of dedicated perceptual representations for *objects, scenes, actions, and words* that leverage the lower-level body-based representations. Later in this section we show results for an approach we've developed that jointly learns about objects and scenes.

Finally, there are perceptual mechanisms that function in an extremely top-down task/goal-oriented mode that recruit object, scene, or body representations in task-specific ways. Understanding the source of the *taskability* is crucial and we present an example from our work on *Visual Routines* that address exactly this issue.

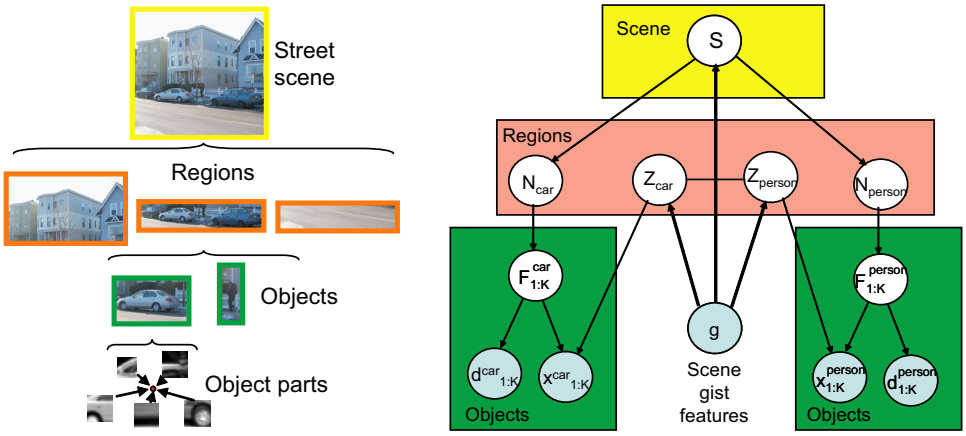
2. Independent loops tightly couple perception to action. Each of the levels just mentioned contains representations that are tightly coupled to the world through some control structures and action. These behavioral loops are independent and hierarchical, e.g. loops that pass through deliberative layers can modulate other loops that pass through reactive layers. Most importantly a behavioral loop can engage different kinds of perceptual modalities and register them through action. We show one such example of how a humanoid robot actively learns a multi-modal (vision & proprioception) model of an object by manipulating it in its hand while looking at it.

3. Bottom-up and top-down processes interact strongly. We know from biology that perceptual processes are characterized not only by bottom-up (or feed-forward) paths where the sensory data go through increasingly sophisticated stages of aggregation/classification, but also top-down (or feedback) paths where the context of the situation, or on-going task exert a very strong suppression or amplification influence on the interpretation of the bottom-up data. For 70% of the neurons in visual areas V1, V2, V3, one-third of their response to a even a simple moving bar is due to *feedback* from higher areas [5]. Weak stimuli in V1 are almost completely suppressed or amplified by feedback from MT. We believe that the use of top-down tasks or context to specialize lower level processes is a crucial feature of biological architectures and is one of the central principles of our architecture. We show an example of how our existing system uses the scene context/gist to help it locate objects.

4. Perceptual representations and processes are re-used for reasoning. We strongly believe that one of the sources of the magic of human cognition is the re-use of perceptual machinery for reasoning. We believe it will be an emergent consequence of the three architectural features mentioned. For example, a possible scenario is that the reflective layer of the perceptual column will be recruited by an abstract task such as adding two numbers or planning a trip, leading the system to project the problem to perceptual representations in the deliberative layer where the procedures synthesized by the "taskability" machinery extracts the perceptual result. thus re-using the perceptual apparatus even for "non-perceptual" problems. See [14] for fMRI evidence that supports this hypothesis.

The perceptual architecture we are proposing is by no means a fleshed out, working system. It is still a framework, albeit one we have a lot of confidence in, based on work already completed (described in the rest of this chapter). A complete description of the perceptual architecture would involve specifying:

1. The *Representations* involved for a variety of tasks; recognizing objects, scenes, and actions for instance.
2. The *Learning and categorization mechanisms* tied to the individual representations.
3. The *Behaviors* (corresponding to the tightly coupled loops) that drive the system to interact with the world, engage the learning mechanisms, and populate the representations.
4. The *Developmental processes* that lead the system to go through stages of learning, i.e. through *conceptual change*
5. How emotions, goals, desires, context (shown in the "control" box of Figure 3.1 *bias* the perceptual processing
6. A *systems level* description of how all of the above results in both reactive behavior as well as what we call thought.



a) Hierarchical scene decomposition

b) A graphical model that relates scenes, regions, and objects

Figure 3.2: Hierarchical model for joint scene and object recognition.

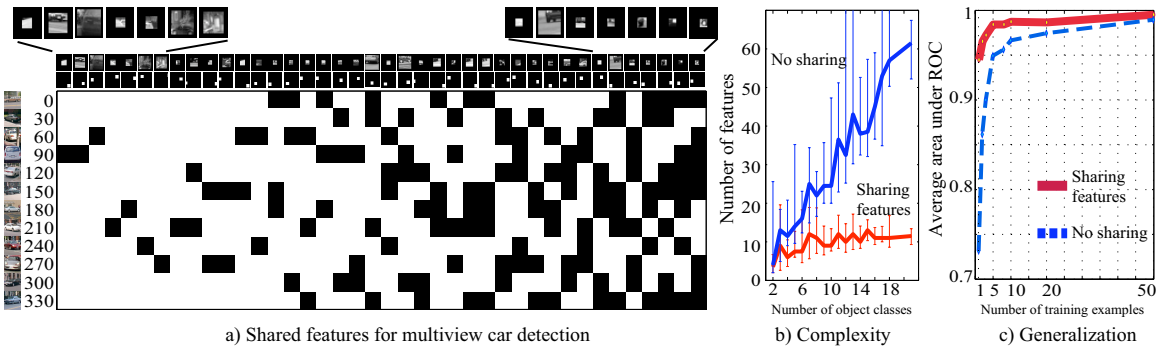
Obviously much work remains to be done before we understand how all the pieces fit together. However, we have already made significant progress on a number of hard issues in perception. In the rest of this section we describe several examples of already completed work each of which touches upon several of the issues mentioned above. In particular:

- **Representations for objects and scenes and top-down influences of scene context**: We show how a system can learn about the *gist* or *context* of a scene and how it uses that representation to inform it about where to look for specific objects or feature of the scene.
- **Goal or task driven perception:** We show how a system uses a sequence of primitive visuospatial operations to locate what a person might be pointing at. The "Visual Routine" for pointing is just one of many visual routines that can be constructed from the same set of base primitives.
- **Active multi-modal object recognition:** We show how a robot manually explores graspable objects and builds a representation that combines visual and proprioceptive information, which it subsequently uses for recognizing the object.

3.1 Integrated model for visual object and scene recognition

Human scene understanding is remarkable: with only a brief glance at an image, an abundance of information is available - spatial layout, scene function, semantic label, identity of main objects in the scene, etc. In traditional computer vision, scene and object recognition are two related visual tasks generally studied separately. However, it is unclear whether is it possible to build robust systems for scene and object recognition based only on local representations and bottom-up architectures. By devising systems that solve these tasks in an integrated fashion, as biological systems must, it is possible to build more efficient and robust recognition systems. An integrated approach provides benefits at several levels:

- Shared features across object classes: significant computational savings can be achieved if different categories share a common set of features. More importantly, jointly trained recognition systems can use similarities between object categories to their advantage by learning features which lead to better generalization. This inter-category regularization is particularly important when few training examples are available, as is common in many vision domains.



a) Shared features for multiview car detection b) Complexity c) Generalization

Figure 3.3: Sharing features for multi-view car detection.

- Contextual relationships: In complex, natural scenes, object recognition systems can be further improved by using contextual knowledge about the objects likely to be found in a given scene, and common spatial relationships between those objects.

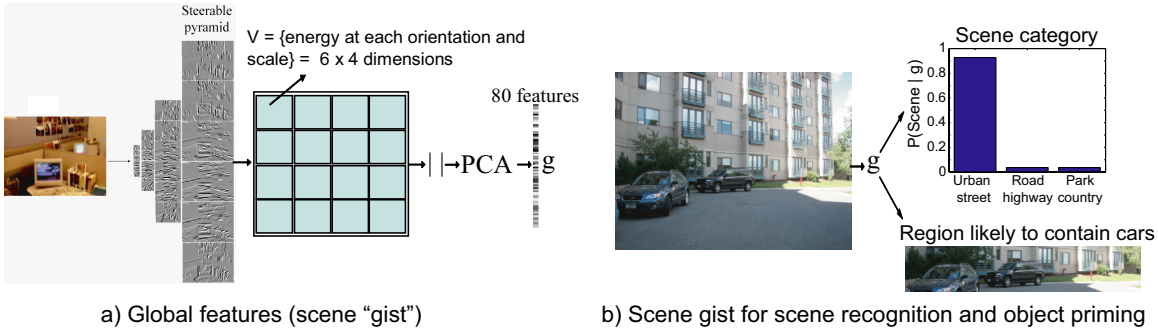
- Scene gist: models of rapid human scene understanding suggest that global image features, independent of object recognition mechanisms, are used to categorize scenes. Global features can be interpreted as a way of summarizing a high dimensional image, as an index that can be used to speed up subsequent processing, suggesting scene category and the objects that are more likely to be present in the scene.

Figure. 3.2 shows the structure of a hierarchical model that integrates scene and multi-class object recognition. The model incorporates visual features at all the levels in the hierarchy providing efficient ”shortcuts” (e.g., scene recognition can be perform even before we have recognized the objects that are present in the image). This is, information does not enter just on the leaves (objects) and then flows bottom-up, instead, image features are used to inform about the scene category and scene layout even before we run the object detector. The next subsections describe the main aspects of each stage:

3.1.1 Shared representations for object recognition

The problem of detecting objects in clutter is often posed as a binary classification task, namely distinguishing between object class and background class, and is typically only concerned with finding a single class of objects. Here we propose a different strategy. There is evidence that area V4 and IT cortex use distributed codes with features that can be shared between different object classes [Rolls97, Pasuphathy02, Tsunoda]. The use of distributed codes have very important advantages for building more robust and efficient algorithms. By learning shared representations between objects, the system needs small training datasets (improved generalization), since many classes share similar features (e.g., both computer screens and posters can both be distinguished from the background by looking for the feature “edges in a rectangular arrangement”), and results in fast classifiers at run time (improved efficiency), since the computation of many of the features can be shared for the different classes.

The learning algorithm we propose [83] (based on Boosting [64]) is an iterative procedure that performs both class clustering and feature selection. At each iteration the algorithm selects one visual feature from a dictionary of features (image fragments [84], Figure. 3.3) containing both class-specific and generic features and selects the group of objects for which that feature is useful. Each training iteration consists of the following steps: first, for each subset of the object classes, find the best single feature that can discriminate the members of the set against the classes outside the set and the background (non-objects). From the set of features from the previous step, select the group of objects and the associated feature that provides the best reduction of the multi-class classification error. Finally, add the selected feature to the multi-class object classifier and re-weight the training samples according to the classification errors produced in the current iteration. The algorithm has the flexibility to select class-specific features if it finds that different objects classes do not share any visual property.



a) Global features (scene "gist") b) Scene gist for scene recognition and object priming

Figure 3.4: Global scene features for scene recognition and object priming.

Figure. 3.3(a) shows a dictionary of features built for the task of multi-view car detection. Here we trained 12 detectors each one tuned to one orientation. Figure. 3.3(a) shows how the features are shared across views (each row is one classifier, and each white entry means that the feature in that column is used by the classifier). The final classifier requires 5 times less training data to achieve the same performance than if we were training a classifier tuned for each view independently. Figure. 3.3(b) shows that as we increase the number of object classes, the number of features used by the system in order to achieve a desired level of performance grows sub-linearly when the features are shared. Figure. 3.3(c) shows that generalization also improves when sharing features, especially when few training samples per category are used.

3.1.2 Scene recognition: the gist of the scene

Contextual influences can arise from different sources of visual information. On the one hand, context can be framed as the relationship between objects. According to this view, scene context is defined as a combination of objects that have been associated over time and are capable of priming each other to facilitate object and scene categorization. To acquire this type of context, the observer must perceive one or more diagnostic objects within the scene (e.g., a bed) and use this knowledge to infer the probable identities and locations of other objects (e.g., a pillow). Over the past decade, research on change blindness has shown that in order to perceive the details of an object, one must attend to it [57, 68]. In light of these results, object-to-object context would be utilized via a serial process that first requires perception of diagnostic objects before inferring associated objects. In theory, this process could take place within an initial glance with attention being able to grasp 3 to 4 objects within a 200 ms window [85, 87].

Alternatively, research has shown that scene context can be built in a holistic fashion, without recognizing individual objects. The semantic category of most real-world scenes can be inferred from their spatial layout only (e.g., an arrangement of basic geometrical forms such as simple geon clusters[4]; the spatial relationships between regions or blobs of particular size and aspect ratio [65]) or from global scene properties [52].

In order to compute this holistic scene representation, the image is first decomposed by a bank of multi-scale oriented filters (Figure. 3.4(a)). The output magnitude of each filter is sub-sampled at very low resolution (4x4 pixels). The final feature vector, used to represent the entire image, is obtained by projecting the low-res filter outputs onto the first 80 principal components computed on a large dataset of natural images. In Figure. 3.2(b), global image features (g) are used to inform the scene category node parallel to object-centered processing. Figure. 3.4(b) shows that the scene gist (g) can be used to recognize the scene and to predict where objects are likely to be in the image. In the example, the gist has selected a region of the image likely to contain cars, providing a task and context driven attentional mechanism. The detector does not need to be evaluated outside of this region, providing a reduction of the computational cost.

3.1.3 Hierarchical model for joint scene and object recognition

The hierarchical model explored here (Figure. 3.2) is a simplification of all the complicated relationships between objects and scenes, and represents a first step [82, 47]. It provides an effective architecture, in which learning and

Figure 3.5: Comparison of results obtained when a view-invariant car detector is evaluated on several images (top row), and the results obtained with the integrated hierarchical model (bottom row). Green bounding boxes denote correct detection and red bounding boxes denote false alarms.

inference is tractable, and improves the performances with respect to standard approaches. In this system two sets of features are used: global features (g), used to represent the context of the scene, and local features (d), used to localize precisely objects in the scene. In this graphical model, both sets of features interact: objects are detected by integrating information from local features (modeling the appearance of the target) and also global features which can be used to select the most likely regions to contain the target using contextual information. Figure. 3.5 show results obtained with a prototype of an integrated system performing object and scene recognition.

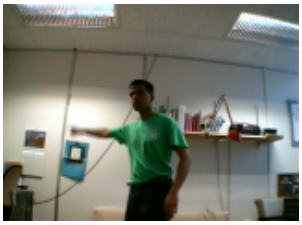
Due to the modular nature of the model, we can add additional sources of information coming from other modalities. Also, temporal information will provide a rich source of information.

3.2 Top-down goal-directed perception: visual routines and attention

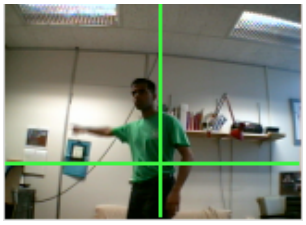
Feed-forward or bottom-up processing accounts for only a fraction of the activity in our visual cortex, the bulk of the activity is from top-down signals flowing back from "higher-level" areas. We know that the human visual system solves an amazing range of problems depending on the goal or task at hand, that provides top-down constraint on what the perceptual systems must do. Without conscious effort, the human visual system finds a place on the table to put down a cup, selects the shortest checkout queue in a grocery store, looks for moving vehicles before we cross a road, and checks to see if the stoplight has turned green. How could the same visual system be "taskable" to solve such a wide variety of visuospatial problems on demand? We have developed a model of visual routines and attention [56] to answer exactly this question.

The key idea is that spatial relations are extracted on demand using sequences of primitive operations called visual routines. There are a fixed set of primitive operations that can be composed in different ways to extract a vast number of spatial relations from the scene. The primitive operations in the visual routines fall into one of three families: operations for moving the focus of attention; operations for establishing certain properties at the focus of attention; and operations for selecting locations. The three families of primitive operations constitute a powerful *language of attention*. That language supports the construction of visual routines for a wide variety of visuospatial tasks.

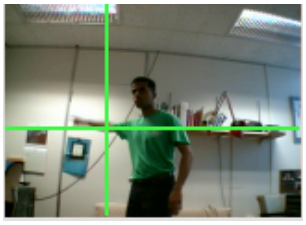
Figure 3.6 shows an example showing the visual routine the system uses to guess what a human is pointing at. In this example the system first locates the human in the scene, then shifts attention to the human, while monitoring the region immediately around the human for a narrow moving object (the arm). Once the arm starts moving up, it is localized, and then a shift of attention brings the arm into the center of attention. Once the arm is in the fovea, its gross orientation is extracted, and that orientation is used to localize a particular area of the room. The selected area is used as a mask to select the most salient object within that area of the room as the candidate object to which the person is pointing at.



(a)

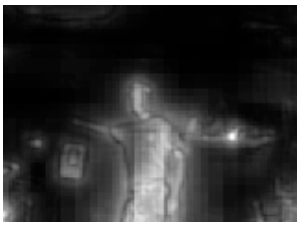


(b) Detect and saccade to the human

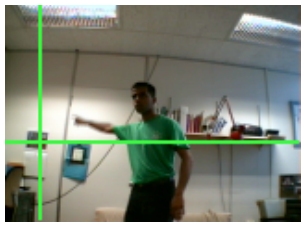


(c) Detect and saccade to the hand

(d) Use the hand orientation to select a portion of the room



(e) Select the most salient blob in the selected area



(f) saccade to the blob

Figure 3.6: An example of a visual routine: The system uses a visual routine to locate what the person might be pointing to and saccades to it.

The sequence of operations in a visual routine is not arbitrary; in the example above note that there are only three classes of operations involved here: (i) *selecting locations*, (ii) *shifting focus* to the selected region, and (iii) *establishing some property of the foveated region* and then starting the cycle all over again. The type of operators chosen each time around may be different but the overall pattern is the same. These 3 classes of operations effectively define "a kind of language of visual attention" from which visual routines for a number of tasks can be composed.

3.3 Learning about objects through action: active multi-modal object recognition

All biological systems are embodied systems, and an important way they have for recognizing and differentiating between objects in the environment is by simply acting on them. Only repeated interactions (play!) with objects can reveal how they move when pushed (e.g. sliding vs rolling), how the size of the object correlates with how much force is required to move it, etc. In a discovery mode, the visual system learns about the consequences of motor acts in terms of such features, and in planning mode the mapping may be inverted to select the motor act that causes a particular change. Learning the perceptual consequences of a motor act, and then selecting a motor act to achieve a certain result, are closely intertwined learning problems, and together are what we mean by "learning to act".

Learning to act is important not only to guide motor behavior but may also be a necessary step for event-interpretation in general, even if the motor system is not involved in any way. For instance, by the age of 6 months children can predict that in a collision with a stationery object, the size of a moving object is related to how far the stationary object moves [37]. This is just one of several things that children appear to learn from experience about their physical environment [74] [75]. What is the source of this knowledge? and how can we build systems that learn to interpret events in the physical world? Computer vision approaches to "object-recognition" and "event-interpretation" have naturally tried to solve this problem in the domain of vision alone. However given that vision does not exist independently of other modalities in biological systems, and knowledge about the world is acquired incrementally in we are taking a somewhat different approach. We assume that *it may be necessary to learn to act on objects first before we can learn to visually interpret more complicated events involving object-object interactions*. One source of evidence in support for this approach comes from the body of work about mirror neurons [62]. These are neurons in motor area F5 of the rhesus monkey that fire when the monkey performs a particular goal-directed action, but which also fire if it just sees another agent perform a similar action. While the mechanisms of this

(a) (b) (c)

Figure 3.7: Motor Exploration populates the View-Transition Object Representation

mapping are far from clear, the fact that the events are mapped to the monkey's existing motor repertoire gives a strong hint that the ability to visually interpret the motor-goal or behavioral/purpose of the action may be helped by the monkey's ability to perform that action (and achieve a similar motor goal) itself.

In prior work [48] we have shown how a humanoid robot that has already learned to saccade, and reach towards points in space with its arm, pushes/pulls an object around in front of it, and learns the effect of its actions on the object, and thereafter uses this knowledge to drive motor-planning. Here we show another piece of work of the same robot building a multi-modal model of a novel object by examining it in it's hand, and thereafter recognizes it using the same kind of visuomotor exploration.

Infants explore novel objects not just with their eyes but simultaneously with their hands as well. It isn't far-fetched to imagine that there might be some highly-specialized and efficient multi-modal object representations for objects that can be grasped and manipulated, different from purely visual representations for non-manipulable objects like buildings or cars. Here we present a simple, novel visuomotor representation called a View-Transition map that relates object views with changes in proprioceptive state while manipulating the object. We show experimental results of how this representation helps a humanoid-robot in self-terminating exploration and then subsequently helps the robot recognizing the object from other objects.

3.3.1 The view-transition map representation

Imagine looking at an object in your hand while rotating it slightly. The transformation results in a new view of the object and a change in the proprioceptive state of your hand. You continue with the exploration by applying new transformations in no particular order, generating observation triplets <start-view, proprioceptive-delta, end-view>. The View-Transition matrix is a way of organizing all these triplets in the form of a 2D map. An entry $M(i, j)$ of the matrix has a proprioceptive vector $\vec{dp}$ that corresponds to the proprioceptive change between View(i) and View(j). Note that this matrix represents the visual and proprioceptive *effects* of the motor-transformation rather than the details (e.g. joint torques) of the transformation itself.

3.3.2 Experiment 1: active object model acquisition

The humanoid robot Babybot (at the Lira lab, Genova Italy) was used for these experiments. It has a 5 DOF head, and a 6 DOF arm. In a typical object exploration trial - the robot would grasp an object handed to it (using reflexive grasping behaviors) and explore the object for about 30 seconds. During this time it would repeat a preset yaw and roll action four times, generating 750 views of the object. For every view, and proprioceptive transition, the system predicted the new view based on the view-transition matrix being built, and compared the actual view with the predicted one. In case of a mismatch - it updates the matrix with the new View pair and proprioceptive vector. While in case of a match it would increment its counter of number of successful predictions and not use the view, transition pair as it contributes nothing new.

The repeated yaw, and tilt action tests to see how effectively the system learns the transitions. On the average the system picked only 12 to 15 views out of 750 as truly unique. Figure 3.9 shows the number of unique views that it acquires, quickly becomes constant, while the ratio of successfully predicted views to views being generated

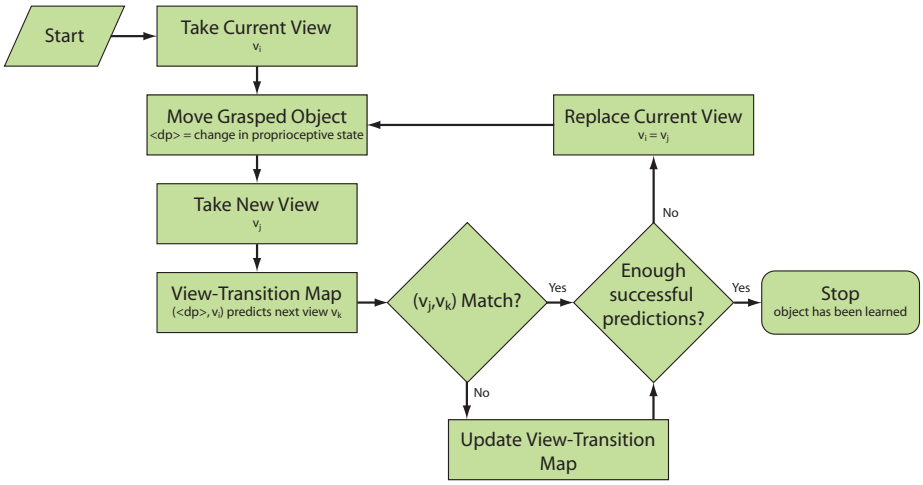


Figure 3.8: Self-Terminating learning of a multi-modal object model; the system knows to stop exploring further, when it isn't surprised anymore

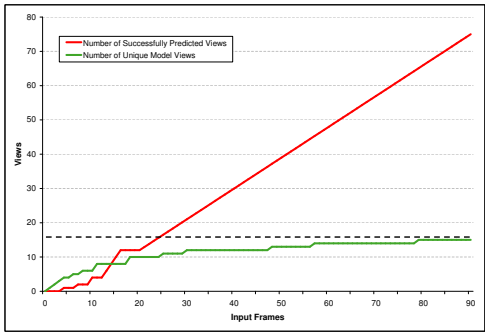


Figure 3.9: Model acquisition through exploration; the number of new views learned quickly converges to a constant while the ratio of successfully predicted views, asymptotes to 1

asymptotes to 1.0. The results shown for the bottle are consistent across all the 11 objects tested. In all cases the system rapidly learns the view transitions and predicts future ones successfully.

3.3.3 Experiment 2: active object recognition using the view-transition matrix

How useful are the view-transition maps for recognizing the objects? More specifically what if anything does the proprioceptive information buy us over and above the views alone? To answer this question the system was trained on 9 Objects types shown in 3.10(a) (2 bottles, 1 stuffed toy, 1 toy gun, 1 ball, a 1 little box, and 3 different configurations of Lego bricks), and 2 control situations (just the empty robot hand going through the same motions).

After learning a View-Transition matrix for each training data-set, the system was then tested with 6 new data-sets corresponding to 6 of the 9 objects it was trained on. Testing involved applying the same sequence of yaw and pitch motions as during exploration, and each triplet $<V_i, V_j, \vec{dp}>$was matched with the view-transition maps of all the 11 classes. Matching was done by using the initial view V_i and proprioceptive transition $\vec{dp}$ to find the best predicted view V_k and matching that against the actual view V_j.

The results for one of the objects is shown in 3.10(b). The x-axis corresponds to each of the 11 object classes (the first two being control) the y-axis shows a match score from 0 to 1 of the test object to the 11 objects types learned. The green bars show the results of using view information alone, while the red bars show the results when the view and proprioceptive information is used. In the sample result shown, the test object clearly belongs to category (e). The visual information alone is sufficient to identify the winning category, but note that the distractor score (the second highest green bar) is quite high. The red-bars on the other hand are much more sharply tuned, with a high-score for the winning category and low scores for all the distractors. This is a fairly typical result showing the advantage of using the View-Transition matrix over, views alone.

To summarize our findings for object recognition

1. Using vision alone is sufficient to identify the most likely category, however it is not very discriminatory
2. The transition map (which relates view pairs with the proprioceptive change) produces more sharply tuned results, emphasizing the winning category while suppressing the distractors.

3.3.4 Conclusion

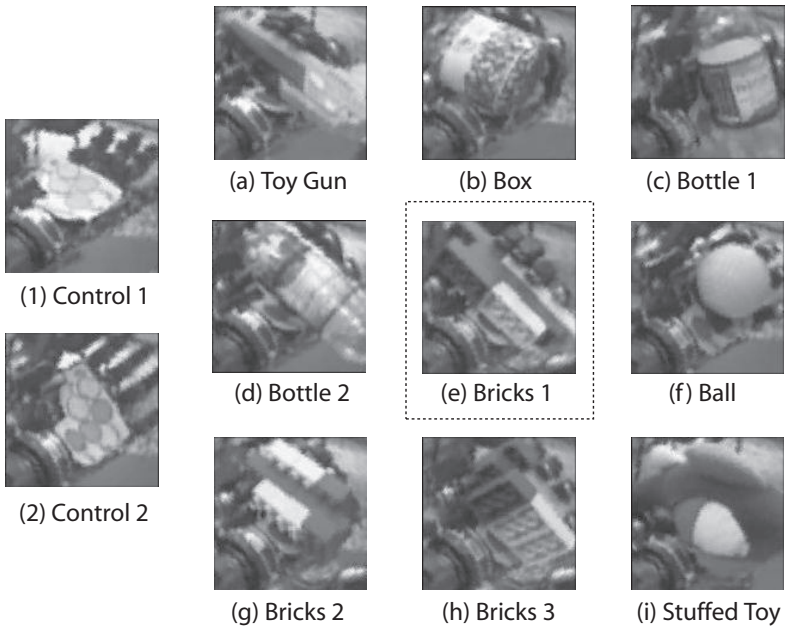
The better results of the view-transition matrix over views alone for object recognition are easily understood when we see that the transition matches are much more restrictive than pure view matches, by bringing *metric 3D information* to bear on the problem.

In conclusion we have proposed a novel, truly "Active" representation for object recognition. The view-transition matrix relates different views of an object by the proprioceptive changes will manipulating it. It has three significant strengths

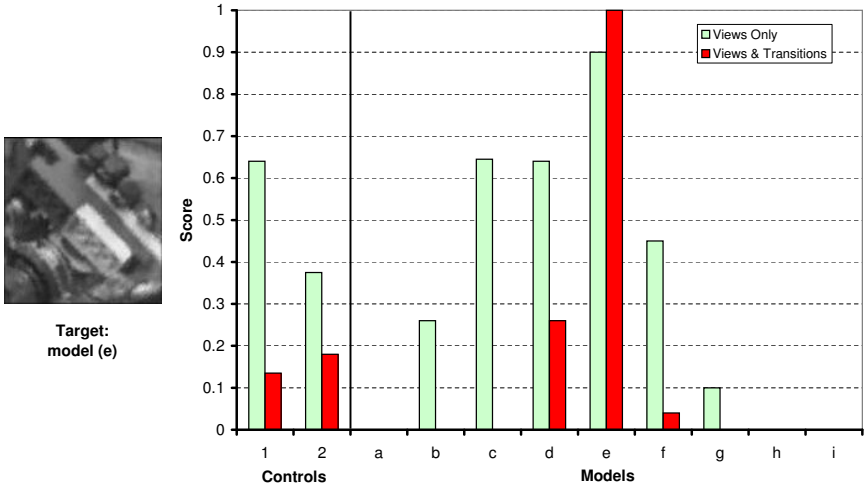
1. It is naturally built up during manual exploration of an object.
2. It makes manual exploration of a single object "self-terminating". The system can know when it is done with exploring a novel object. It can stop exploring once the VT matrix being built up, correctly predicts most views (i.e. the system is not "surprised" anymore)
3. It makes object recognition more precise over a purely vision or view-based strategy by bringing 3D metric information to bear on the problem.

3.4 Summary

What role does perception play in human-like cognition, and how? We deviate significantly from traditional production-rule systems where perception is a "front-end" module whose job is to deliver symbols about the state of the world to a symbolic reasoning system. Instead, in our architecture non-symbolic perceptual representations and processes are central to reasoning. Key aspects of our architecture for perception are:



(a)



(b)

Figure 3.10: (a) Shows the Object classes used for training, the first two are control conditions (b) shows the results of Object Recognition for class 3 the x-axis shows the object categories, y-axis shows the match using views alone (green) and including the proprioceptive transition information (red). the bricks are recognized as the most likely category by both types of matches, however the view-transition matrix matches (in red) are more sharply tuned.

- Three qualitatively different kinds of perceptual representations and processes.
- Independent, hierarchical loops that tightly couple perception to action.
- Interaction of bottom-up and top-down processes.
- Re-use of perceptual representations and processes for reasoning.

To illustrate these key features of our architecture, we presented a sample of results from work already done.

Learning object and scene models and top-down influences of scene context: We showed how a system learns about the *gist* or *context* of a scene and how it uses that representation to inform it about where to look for specific objects or feature of the scene.

Taskability - Goal or task driven perception: We showed how a system uses a sequence of primitive visuospatial operations to locate what a person might be pointing at. The "Visual Routine" for pointing is just one of many visual routines that can be constructed from the same set of base primitives.

Active, multi-modal, object recognition: we showed how a robot manually explores graspable objects and builds a representation that combines visual and proprioceptive information, which it subsequently uses for recognizing the object.

Chapter 4

Brain-Inspired Actuation

The CHIP actuation architecture is motivated by a number of fundamental features of the human actuation system, by which we mean the system that translates high level goals involving controlled postural maintenance or movement of the body or of environmental elements. First, it is clear that in general motor skills are strongly degraded by loss of sensory feedback [26]. For practical purposes, the motor system is not intended to operate in a strictly feedforward manner. This is especially reasonable for predatory animals (such as humans) that operate in open and populated environments within which the dynamics and disturbances are naturally unpredictable. Sensorimotor control loops are the rule. Second, it generally takes humans six to eight years of rich physical activity to become proficient at voluntary motor skills. Moreover, gross and fine motor skills may be refined and lost with substantial independence. While it is likely that this is related in part to the late maturation of high speed corticospinal and cerebellar tracts, it is also quite likely that it takes many years to develop a rich and refined repertoire of sensorimotor commands. Third, motor skills may improve and decline somewhat independently of many other cognitive skills. Yet, motor skills decline significantly in Alzheimer's disease and other cortical dementias, strokes in the speech actuation area of cortex (Broca's area) interfere with certain types of comprehension, basal ganglionic lesions are associated with "subcortical" dementia, and cerebellar dysfunction interferes with a number of cognitive processes. These observations indicate that fundamentally the actuation system is richly interconnected with sensory inputs, that human levels of motor skill can be reproduced by an extremely rich, refined and accessible cache of routines that can be invoked according to current and anticipated sensory contexts, and that actuation system is designed to be closely integrated with and to inform the other spheres of cognition. Thus, we have viewed our task as that of developing a highly efficient, hierarchical mechanism for establishing, selecting and executing a rich collection of sensorimotor routines in a manner that is most consistent with the architecture of the primate sensorimotor control system.

4.1 Specifically relevant neuroanatomy

The physical organization and basic physiology of the human motor control system provides a very useful guide for the design of an intelligent actuation system. Figure 4.1 serves as a central cartoon. The human system consists of a hierarchy of control loops that enable responsibility for control to be distributed likely for three principal reasons. A. there is a survival advantage for those animals whose lower levels can assist it in defending itself, escaping, feeding and reproducing when higher levels have been damaged. B. coding phylogenetically earlier/simpler systems are retained and are built upon when further skills become needed. C. if properly distributed and decoupled, learning and adaptation can proceed independently, simultaneously and often with minimal compromise of overall system stability. Thus, it is likely that the spinal cord circuits are sufficient to organize gross movements in terms of cooperating synergies of muscles. Simple but quite effective defensive and propulsive movements can be organized by these circuits . Longer but very simple feedback loops through brainstem cerebellum and motor cortex that drive such spinal synergies appear to be able to produce highly effective upright balance and locomotion. Further loops that include basal ganglia are very likely to be responsible for a range of legged behaviors Visuomotor control is likely driven by similar loops that that traverse premotor cortex. Finally, some humans develop high levels of skill with their legs and feet from the soccer player, ballerina, floor gymnast and even bibrachial amputees who write

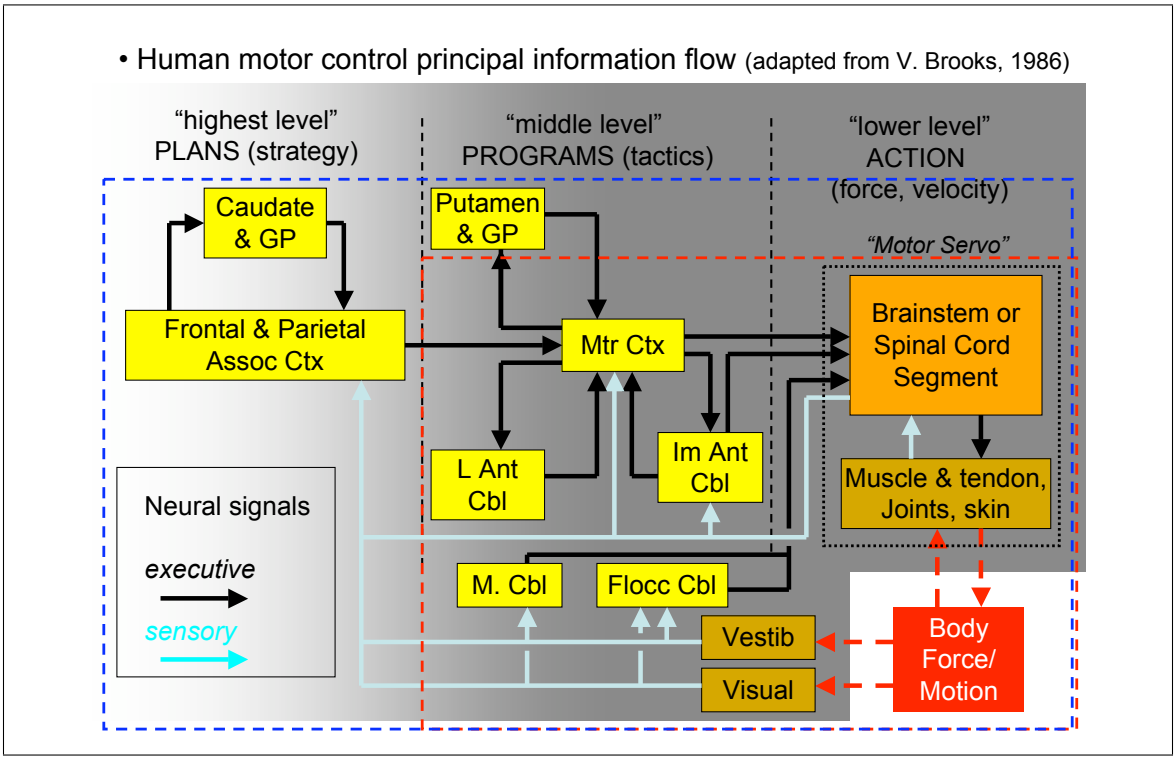


Figure 4.1: Gross functional neuroanatomy of sensorimotor actuation system. Control loops are ubiquitous. The continuous or impedance control level represented by the RIPID control system in Figure 2 is outlined in red. The Discrete Time Epoch (DTE) control level is outlined in blue.

with their feet. These skills are developed after the previously listed skills, and are compromised by injury to any of the lower level loops. This confirms the hierarchical structure of the control apparatus and strongly suggests that the control of upper and lower extremities is fundamentally similar.

4.2 Modeled and abstracted architecture of the human motor control system

4.2.1 Continuous and discrete time control levels

In light of the above observations, the work of three of the BICA investigators Williams, Hofmann and Massaquoi in developing in parallel detailed neuroanatomic ("RIPID based") and biomorphic AI (Locomotion Task Executive) versions of the human locomotor control system (Figure 4.3) appears to have direct relevance to the manual manipulation problem. Both control systems are hierarchical feedback control schemes that produce realistic dynamic simulations of human walking and are in the process of being applied to physical robots. In the Locomotion Task Executive, the lowest level (the model based controller) uses an internal dynamic model of the limb and foot to linearize the control. This corresponds to the action of lower spinomuscular reflex loops and muscle synergies together with stabilized trans cerebrocerebellar feedback loops in Jo and Massaquoi's architecture [35]. In the latter, the cerebellum can provide linear (proportional) scaling, approximate temporal (mathematical) differentiation, and approximate temporal (mathematical) integration. These are the core building blocks of linear dynamic controllers. The spinal and motor cortical circuits are responsible for distributing the cerebellar control signals to the muscles in the most useful ratios (synergies). The cerebellum is viewed an enormous repository for stored gain settings that can be selected dynamically according to current control demands. It seems that in nature, gain scheduling substitutes for the internal dynamics models used by the Locomotion Task Executive architecture. In both architectures, then, the lowest level circuits enable context-sensitive, piecewise proportional integral derivative (PID) continuous time control of the limb. We will refer to this lowest level as the "PID-impedance control level". Massaquoi [39, 41]

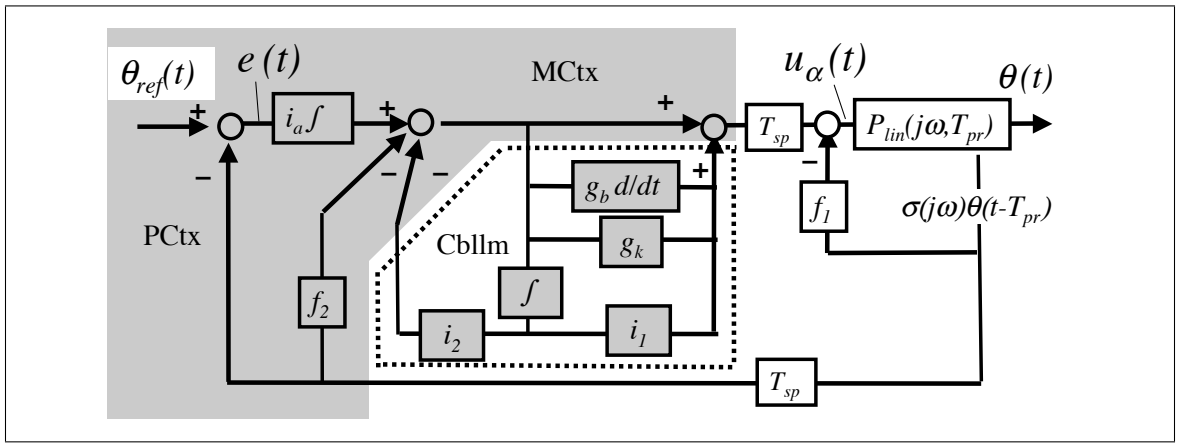


Figure 4.2: The base Recurrent Integrator Proportional Integrator Derivative (RIPID) cerebro-cerebellar control model manages continous time, impedance control of limbs. MCtx: motor cortex; PCtx: parietal cortex; Theta_ref(t) reference trajectory, e(t) tracking error signal; T_pr, T_sp: peripheral and spinal signal transmission delays, respectively. Cerebellar proportional, derivative and integrator processing is indicated within the dotted boundary. The internal negative feedback traversing block i2 is a "recurrent integrator" that is proposed to stabilize the long feedback loops and to account for "cerebellar outflow tremor" when lesioned.

has shown that such context-sensitive (i.e.gainscheduled) PID can be used successfully to control arm reaching movements as well.

The next higher level in the locomotion task executive architecture will be referred to as the "discrete time epoch (DTE) control level". It consists of a discrete time state-based controller that receives from above a "State Control Plan (SCP)" specifying a series of trajectory objectives in the form of (space, time) intended via points (or regions) within the workspace and provides the context-specific nominal trajectory and parameter settings that should enable the PID-impedance control level to implement the specified trajectory segment (or maintain the posture). The controller monitors the ongoing body motion of the body and determines whether the current control settings are adequate or inadequate to achieve the target or goal of the current segment. If not, it determines whether alternative settings within pre-specified limits would be successful. If so, these settings are instantiated. If not, the next highest level is informed of impending failure.

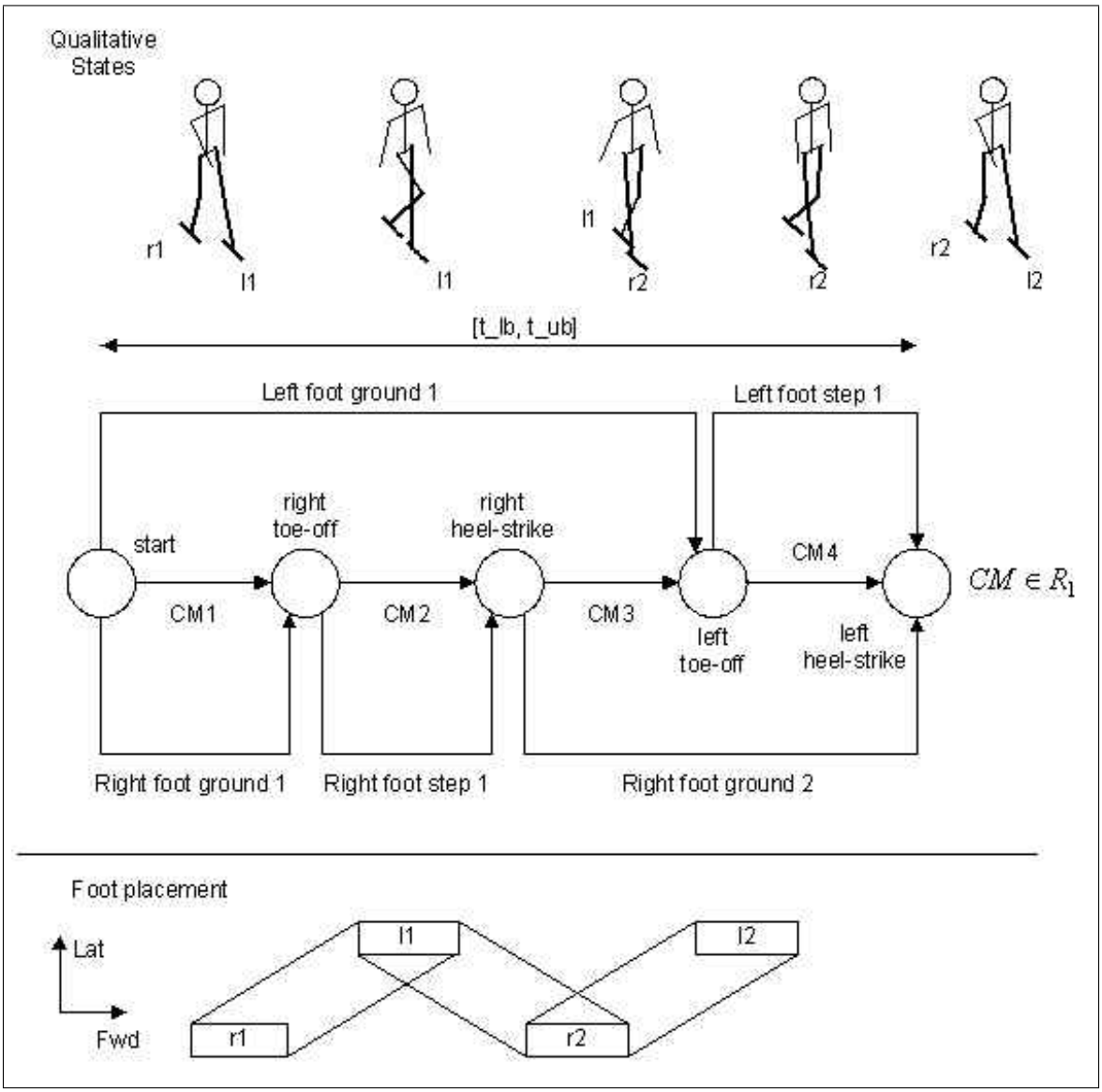


Figure 4.3: Example qualitative state plan for walking gait cycle. Circles represent events, and horizontal arrows between events represent activities. Activities may have associated state space constraints, such as the goal region constraint , which specifies a goal for CM position and velocity. Foot placement constraints are indicated at the bottom; for example, rectangle r1 represents constraints on the first right foot position on the ground, and rectangle l1 on the first left foot position. The lines between the rectangles define the polygon of support when in double support.

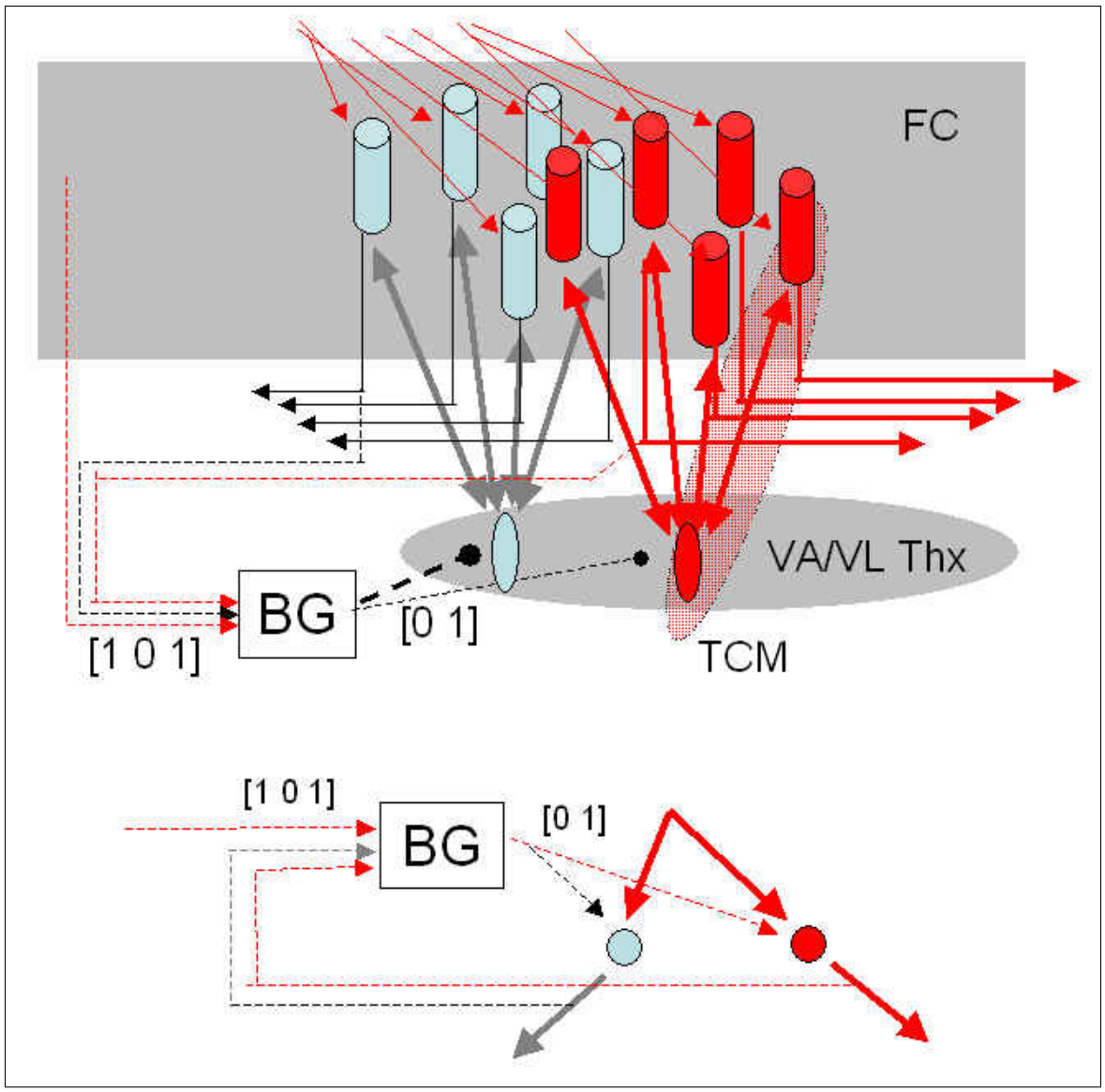


Figure 4.4: Schematic representation of basal ganglionic (BG) gating of frontocortical (FC) columnar assemblies. (Top) BG module receives context input as binary patterns of active or inactive channels (red and black, thin dashed) based on state of some parts of cortex. Output (black, dashed) is inhibitory to VA/VL nuclei of thalamus (Thx). Weak inhibition allows assemblies of FC thalamocortical reverberatory modules (TCM) to respond to diffuse input from other cortical areas (descending red). Conversely, BG inhibition can block activation of other columnar assemblies. (Bottom) Compact representation of circuitry at top as used in following figure. Activated FC assemblies represented by red disc, inactive by pale blue. BG therefore acts as a context-dependent selector switch on FC activity representable as mapping between binary context and control vectors.

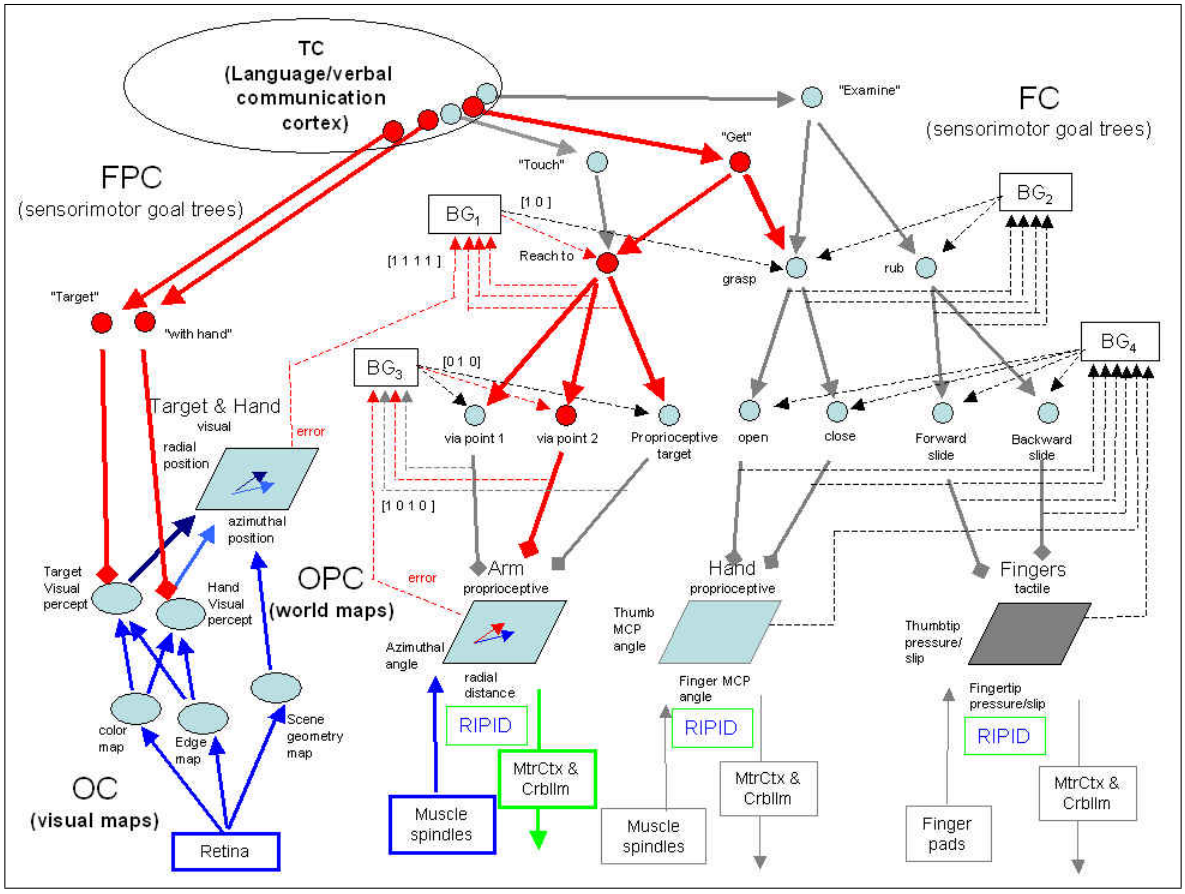


Figure 4.5: Schmatic use of cortico-basal ganglionic circuits to implement hierarchical cache-driven sensorimotor control loops (stored sensorimotor 'program') for manual manipulation. Salient features are: neuronal assemblies that can be active or inactive as gated by the BG through thalamus; thalamocorticallogical hierarchy of neuronal assemblies in frontal cortex; basal ganglionic modules (BGi) that enable=1/disable=0 neural assemblies according to input contexts represented functionally by binary vectors (e.g. [1 0 0 1]) where 1 represents nonzero, 0 represents zero/inactive; sensory maps that register deviations of actual (sensed) state (blue) from intended state (red); system may be activated by verbally articulated goals (e.g. "get object") associated with assemblies in language areas, and may contain nonverbally articulated goals (e.g. via point 1, via point 2, close, slide forward); ultimately intent is represented as intended sensory states in parietal cortex (PC) that are matched against actual (sensed) states; nonzero error signals indicate that process has not been completed. Thus, there is structural uniformity across the hierarchy, the motor "program" "resides" as a distributed representation. Memory storage and execution circuits are identical, obviating "fetch" operations. Direct associative "lookup" from language and visual areas obviates "search" operations. Sensorimotor loops such as visuomotor loops and proprioceptive-motor loops may be nested. Subprograms such as "grasp" may be shared by having dual pointers from above. Context and control vectors update every 80 to 100 ms in vivo. Aside from this minimum interval limitation, separate parts of system may operate with various temporal relationships ranging from synchronous to serial depending upon coding within BG. At the moment depicted here, the hand is being controlled toward the second via point en route to the targeted object. Hand opening will begin at some subsequent time and will be followed by hand closure. Rubbing of the object will NOT occur unless verbal command to examine object is received. This could be one of thousands of cached implementations having different via points and grasping strategies (e.g. hand rotation could be included).

The operation of the locomotion task executive system's discrete time state-based controller in implementing the SCP is highly analogous to the operation of both spinal pattern generators and the interaction between frontal cortex, basal ganglia and cerebellum proposed by Massaquoi and Mao [40]. Jo and Massaquoi [34] have shown that very simple pulse-like commands into the PID-impedance control level can generate basic locomotor patterns seen in humans. Similar, but much more flexible control appears to be provided by the basal ganglia and frontal cortex. According to this view, the core function of the basal ganglionic loops is to gate (enable or disable) the activation of frontal circuits according to the current state of the cerebral cortex (Figure 4.4). Because this process can be iterated, the net effect is to produce an automated procedure or program of frontal cortical activations (Figure 4.5). In the SCP transitions may occur at arbitrarily small intervals. In the human brain the analogous transitions may up to every 80 to 100 ms. This corresponds to "mu" range (analogous to alpha rhythm in motor areas) cortical frequencies in the cerebral EEG. The minimum interval is due at least in part to the typical cortico-basal ganglionic loop timebut is also consistent with the natural resonant frequency of the large layer 5 cells in cortex that can recruit cortical columnar assemblies. Thus, in the brain, every 80 ms interval then serves potentially as a new motor control epoch during which a new intermediate or final trajectory target, as represented by a cerebral cortical modules, and new controller gains as represented in cerebellar modules, can be specified.

In both the locomotion task executive architecture and apparently in nature, the discrete time control specified by the DTE level is interpolated by the PID-impedance effectively implementing discrete time-to-continuous time conversion. As in computer systems, the representation of continuous trajectories by discrete time samples (targets) and piecewise constant parameter settings affords potentially a great reduction in memory requirements for storing motor procedures. Because the state of the cerebral cortex typically changes with sensory input, natural motor procedures can be interactive with the environment and can thereby yield sensorimotor procedures. Because the frontal cortical representations can themselves be organized hierarchically, the resulting sensorimotor procedures can exhibit nesting. In particular, should the result of an action result in sensory signals that indicate impending failure of an action to achieve its desired goal, a discrete transition can be made to another procedure to manage the failure. For example, rapid transition of the center of mass beyond the base of support can trigger a step. Or slippage of an object within a grasp can trigger a regrasp. Similarly, at a higher level, slippage of the object from the grasp could invoke an entirely different task. In the locomotion task executive architecture, task-level planning is represented by a Qualitative State Plan (Figure 4.6 and below). Here a sequence of gross movement goals are represented and are reconciled with the constraints provided by the body and environment. This results in a State Control Plan that is passed to the DTE controller. In principle, both the QSP and SCP can be represented within the tree structure of Figure 4.5. Thus, in both the locomotion task executive architecture and apparently in nature, a multi-level hierarchical discrete-time, continuous-time system enables implementation of flexible interactive routines that could be applied to all types of sensorimotor control tasks. The general idea is already used by many robot control systems. However, the particular tradeoffs made and the strong potential for implementation with parallel hardware has not been exploited heretofore.

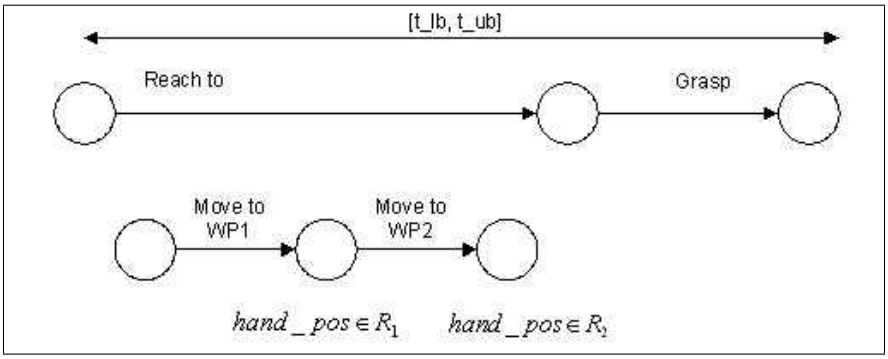


Figure 4.6: Qualitative State Plan for reaching and grasping represented in hierarchical form suitable for implementation by a cortico-basalganglionic network.

4.2.2 Massive memory cache

The artificial and natural architectures described above have the particularly attractive feature that they are general purpose and highly flexible so they easily take advantage of massive storage of simply coded routines organized in hierarchical, tree-like structures. A central challenge is then the establishment of such hierarchically organized caches. Here the work of Robertson in experiential acquisition, refinement and organization of command strings is particularly relevant. The general scheme, which we refer to as hierarchical event string caching is to develop a hierarchical, multi-resolution representation of the sensory image - actuation command chains that can be quickly invoked to achieve a goal. In particular, once the chains are represented as tree-like nodes within hierarchical neural networks (Figure 4.5) navigation down and up the chain can be described as simple context-driven enablements/disablements of the type readily afforded by fronto basal ganglionic circuitry as outlined above (Figure 4.4).

On the other hand, development of the hierarchical representation can be afforded by simple dynamics between the levels as can be argued to be easily consistent with inter-areal cortical interactions.

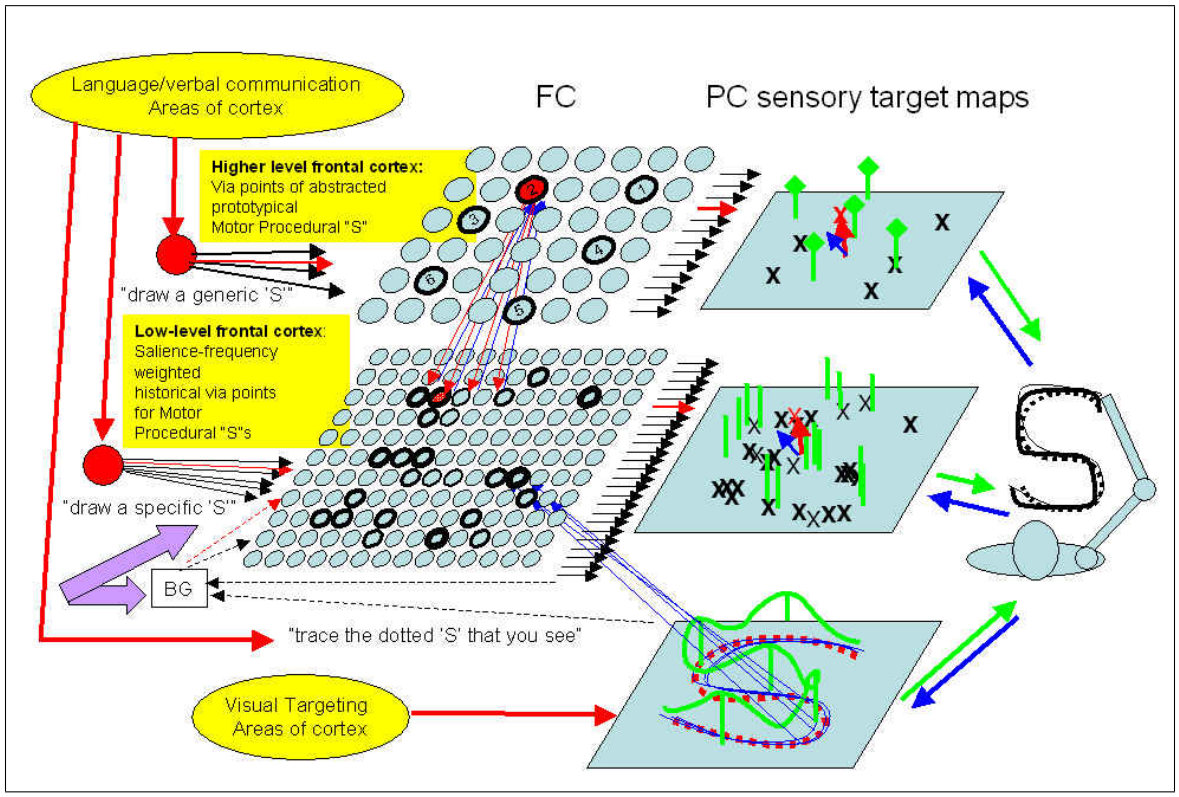


Figure 4.7: Scheme for establishing hierarchical cache via points through practice. Example given is developing a hierarchical sensorimotor programs for drawing the letter S.

Figure 4.7 outlines the implementation of Robertson's hierarchical event string caching algorithm using basal ganglia-like circuitry. The first stage, the exploration stage, installs sensory image – actuation command pairs in first tier visual (Figure 4.7, bottom) and frontal and somatosensory cortical areas (Figure 4.7 middle). The actuation command is invoked by sensory context-dependent basal ganglionic enablement of the next frontal node. Successful outcomes enhance storage of these pairs, while unsuccessful outcomes link them only weakly. This type of reinforcement learning appears to be consistent with the role of dopamine and system limbic activation in basal ganglia and possibly frontal cortex in strengthening or weakening connections in both areas. These sensory context - actuation command pair linkages can be built up over time with experience/practice, until the configuration space is well populated with useful sensory-command strings. Those motions that are most successful behaviorally as marked by dopamine release supervised according any behavioral criterion specified elsewhere (e.g. fastest, smoothest, fewest

obstacles, shortest etc.) will have the strongest sensory-action linkages. Next, Second tier cortical areas "observe" the activations in the first tier areas and identify statistical regularities. In particular, sensory images that are frequently encountered because they are common to a large number of first tier sensorimotor command chains develop a strong neural representation in the second tier areas. As a result, the second tier (Figure 4.7, top) develops an abstracted, coarser representation of the successful command strings. As in Robertson's algorithm, less successful sensory-action linkages are presumably designed to fade.

It is readily appreciated that if neural representations within this network are allowed to compete Using "winner-take-all" mechanisms attributed to basal ganglia and areas of cerebral Cortex that eventually, direct access to the most context-appropriate sensorimotor strings can be fostered. Because of the comparative shallowness and breadth of the treelike representation, selection may be very fast despite massive numbers of cached routines. In this way QSP and CSPs can be acquired and executed using fronto-basal ganglionic type circuitry.

4.3 Comparison with Traditional Approaches

Traditional robotic actuation controllers rely on a two-step process. The first step is trajectory planning, and the second is trajectory tracking. The goal of the first step is to produce a reference trajectory that accomplishes the required action. This is normally a computationally intensive step that uses numerical optimization techniques, and takes into account dynamic and kinematic constraints, and a variety of optimality criteria (Popovic brothers, Kino-dynamic planning, Kuffner). The goal of the second step is to follow the reference trajectory as closely as possible. This is normally accomplished by a high-gain (high-impedance) tracking controller.

There are two significant disadvantages to this approach. First, using high-impedance, non-compliant tracking can be dangerous in that the robot appendages are very stiff. Thus, if there is a disturbance, or they collide with an unforeseen obstacle, they may cause damage to themselves, or to their environment. Second, blindly following a single reference trajectory is not always the best thing to do, again, when there are disturbances. Typically, there are many trajectories that satisfy the overall task requirements. The control system should be smart enough to know when it is on one of these trajectories, and when it isn't. If it isn't, then it knows immediately that the task won't succeed, and that it should give up.

For this reason, our parallel approach of storing many or all trajectories that satisfy a task is advantageous. Because the trajectories are pre-computed, there are no concerns of heavy computation in real time. An efficient mechanism for retrieving an appropriate trajectory for a particular situation is all that is needed.

4.4 Possible implementation with current hardware

The architecture described places most of the computation burden upon finding neareast neighbor matches in a high dimensional space. While this computational burden is a poor fit for conventional Von Neumann machines it has a number of attractive features:

1. The approach is very robust to small pertubations in the perceptual space. This robustness derives from the very high dimensional spaces involved. High fan-in is a defining feature of neuronal circuits and is also a feature of our approach.

2. The approach is very highly parallelizable. The basic computational elements can operate in a loosely coupled way and the computing elements are uniform so a large number of them can be replicated compactly.

Current implementation techniques using, for example, kd-trees, scale poorly limiting the size of the problem that can be managed on traditional hardware. Some parallelism is available with the use of vector processors such as are available in modern graphics processing chips. Our current implementation runs purely in software and permits testing with small sized problems. Our first step will probably involve algorithmic advances and the use of graphics/game processors to achieve performance suitable for a larger problem domain. Beyond that the real opportunities for perfromance and scale come from building special purpose highly parallelized circuit that perform the cache look operations in parallel. We can easily build such a system by using current generation FGPAs, such as tha Xilinx Virtex II, using FGPA development boards.

4.5 Compatibility of Qualitative State Plan with CHIP actuation architecture

We seek to show how a robot can be guided so that it accomplishes a specified task, such as walking on a set of irregularly placed stones, walking to a soccer ball in time to kick it, or grasping and manipulating an object.

For most practical applications, a precise specification of state and temporal goals is not necessary. Rather, a loose, flexible specification, in terms of state space regions and temporal ranges, is preferable in that it admits a wider set of possible solutions. This may be exploited, for example, to improve optimality or to adapt to disturbances.

An example state space goal is for the robot's center of mass position to be within a particular region, or for its hand to be in a particular region. An example temporal goal is that such a state space goal be achieved after 5 seconds, but before 6.

We define a qualitative state as an abstract constraint on desired position, velocity, and temporal behavior of the robot. All quantitative states corresponding to a qualitative state share important characteristics which define their membership in the qualitative state. For example, all the quantitative states satisfy the constraints defined for the qualitative state. In particular, such constraints may include model parameters that change significantly from one qualitative state to the next. For example, a bipedal robot with two feet on the ground can move very differently from one with only one foot on the ground. This difference is captured using distinct qualitative states: one for double support, and one for single support.

Accomplishing a goal may require transitioning through a sequence of qualitative states. Thus, a sequence of qualitative states represents intermediate goals that lead to the final overall goal. Such a sequence forms a qualitative state plan. For example, reaching a goal location may require a bipedal robot to take a sequence of steps. Such steps represent transitions through a sequence of fundamentally different qualitative states, defined by which feet are in contact with the ground.

An example qualitative state plan (QSP) for walking, is shown in Figure 4.3. The QSP has a set of activities representing constraints on desired evolution of state variables.

Activities are indicated by horizontal arrows in Figure 4.1, and are arranged in rows corresponding to their associated state variables. Every activity starts and ends with an event, represented by a circle in Figure 4.3. Events in this plan relate to behavior of the stepping foot. Thus, a toe-off event represents the stepping foot lifting off the ground, and a heel-strike event represents the stepping foot landing on the ground. Events define the boundaries of qualitative states. Thus, the right toe off event defines the end of the first qualitative state (double support), and the beginning of the second qualitative state (left single support).

The qualitative state plan in Figure 4.3 has a temporal constraint between the start and finish events (between the beginnings of the first and fifth qualitative states). This constraint specifies a lower and upper bound, , on the time between these events. Such temporal constraints are useful for specifying bounds on tasks consisting of sequences of qualitative states. The temporal constraint in Figure 4.3 is a constraint on the time to complete the gait cycle, and thus, can be used to specify walking speed.

In addition to temporal constraints, qualitative state plans include state space constraints. These are associated with activities, and are specified as regions in position/velocity state space. Such regions can be used to specify required initial and goal regions, as shown in Figure 4.3. If an initial region is specified for an activity, then the trajectory must be within this initial region, in order for the activity to begin. In Figure 4.3, the goal region constraint represents the requirement that the CM trajectory must be in region for the CM movement activity to finish successfully.

Figure 4.6 shows a QSP for the get object task depicted in Figure 4.3. In this case, the QSP is arranged hierarchically, with the move to waypoint activities being sub-activities of the reach to activity.

QSP's such as the ones shown in Figures 4.1 and 4.3 are interpreted by a task executive. The task executive is responsible for controlling the plant in such a way that all constraints specified in the QSP are satisfied. Thus, at the beginning of an activity the task executive checks that the initial conditions are satisfied. If so, it issues control commands to the plant. If not, the plan has failed. The task executive monitors plant state, updating control commands to keep the plant on track. It also monitors the plant state to check if the activity goals have been reached. If so, the task executive transitions to a new activity. When all activities have been successfully executed, the QSP

has been executed. If any activity execution fails, then QSP execution has failed.

Because of the hierarchical representation, it can be appreciated that cortico-basal ganglionic-like circuits in Figure 4.5 could implement a task executive that sequences the robot through the qualitative states in the QSP of Figure 4.2. By checking context (current qualitative state), and by issuing sequencing commands, the basal ganglionic circuits control transition to new activities and qualitative states. Within each qualitative state, the cortico-basal ganglionic-like circuits also could enable or inhibit control laws corresponding to the state. These control laws are executed by the motor cortex, aided by sensor information from the muscle spindles. The control laws keep the motion trajectories within the bounds specified for the qualitative state.

Chapter 5

The Chip Reasoning and Decision Making Architecture

5.1 Revisiting the scenario

We begin this section by revisiting an extended version of the scenario presented in the overview section to provide a backdrop for explicating the CHIP architecture's approach to human decision making.

> George is about to pay his bills and is trying to find his favorite silver pen. The pen was given to him as a present; he is sentimentally attached to it and enjoys its feel. He knows he could get another pen, or use a pencil, but right now he's a little obsessed with finding this particular pen. He can picture putting it down on the kitchen table, but doesn't see it there. He thinks about where else he usually leaves his pen and decides to look in his study, but he doesn't find it there. George thinks about calling up to his wife to ask if she's seen the pen, but he doesn't want to hear a lecture about how he should keep better track of his stuff, so he decides not. At the same time acknowledging to himself that he really ought to keep better track of his stuff.
>
> Noting that he didn't search thoroughly the first time, he returns to search the kitchen again. Walking back to the kitchen he notices a shiny object out of the corner of his eye. While turning to focus on it, he hears a noise and automatically turns his head towards it; he sees the cat running away. He turns back to the shiny object and sees that it is his pen. George makes a mental note to kill that cat sometime soon

5.1.1 Observations

Human decision making is extremely complicated, even in the most mundane situations. We pursue our goals by finding ways to translate them into actions while simultaneously reacting to our environments. We observe, we react, we deliberate, and we reflect on what we've done. We monitor our actions to make sure that they have the effect we intended. We maintain complicated internal state, including memories, emotions, desires, a sense of place and a task context. We are capable of doing all of this at the same time.

We know an incredible amount about the structure of the everyday world and manage the complexity of the world by exploiting regularities to abstract away details not relevant to our goals. We also use this regularity to imagine what might happen were we to take an action. We are self-aware in the sense that we can remember regularities about our internal states: we know what we just deliberated about, what our general tendencies are, and where our strengths and weaknesses lie. We are socially aware; we consider how our actions will affect others and how others will affect us. We have internal conversations with ourselves. We harbor grudges.

The short story above illustrates many of these points:

- *George is trying to find his pen*: George has the **goal** of locating his pen and in the story he selects several **plans** for doing so.
- *He knows he could get another pen, or use a pencil*: George has a large **web of knowledge** relating to mundane things like pens. He knows how to recognize a pen. He knows what pens are for; He knows how to use a pen. He has **mental images** of specific pens and this "imagery" includes both pictorial and tactile modalities. He knows that pens are used with paper, that people can own a pen.

George knows that what he needs is a writing instrument and that both pens and pencils are parts of this semantic category; George could use this **semantic knowledge** to form alternative plans that involve using other writing instruments. The concept of "pen" is wired into all of George's mental apparatus; a result of interacting with the world and **identifying regularities** across experiences.

- *but he's a little obsessed with finding that particular pen right now*. In choosing how he's going to satisfy his top level goal of doing the bills, George is concerned with many other issues, such as his enjoyment of using this particular pen. Emotions figure centrally in how he prioritizes his goals, and selects plans and actions to achieve them.

- *He can picture putting it down on the kitchen table*: George has memories of past experiences and can "replay them in his mind's eye". In particular, he can retrieve a recent memory of an episode in which he got up to answer the phone and put the pen on the kitchen table. He can use his perceptual subsystems to "see" where he put the pen and to guide his current search for it.

- *but he doesn't see it there*: George uses perception to **monitor the execution of his plans**. Sometimes his **goals influence his perceptions**, as in this case where he will ignore most things other than his pen, even when they're right in front of him.

- *He thinks about where else he usually puts the pen*: George can **reason about his own behaviors**. In this case, one part of his mind detects a failure of the current plan. He can then choose to pursue a different strategy. To do this he must have **internal models** of his own normative behavior. These models are built by extracting regularities from past experiences, including internal plans and expectations.

- *and decides to look in the study*: George has an **abstract mental map of the physical space** that he's acting in. He breaks this up into qualitative chunks (e.g. rooms) and trajectories between them.

- *He decides to check the kitchen again, noting that he hadn't really looked that thoroughly the first time*: George can **reflect** on how thoroughly he had carried out a previous plan and revise it if necessary.

- *Walking back to the kitchen, he notices a shiny object out of the corner of his eye. While turning to focus on it,*: George's perceptual system is primed now to respond to perceptual evidence that relates to his goal of seeing the pen. His deliberative layer then creates a new subgoal to validate or invalidate the observed evidence and this temporarily halts the subgoal of returning to the kitchen, as the new subgoal might satisfy the high-level goal of finding the pen.

- *he hears a noise and automatically jerks around towards it*: George has **wired in and acquired reactions** that rapidly connect perception to action. In this case, the reaction is probably one whose original purpose was to avoid fast moving predators and causes an action before deliberation can override it.

- *He sees the cat running away*: George knows that cats like to play with small shiny objects like pens. He reasons that maybe the cat stole the pen; George can picture the cat playing with a pen in his mind's eye, even though he may not have ever witnessed this cat actually playing with his pen. George is able to create **internal simulations**.

- *George reminds himself to kill the cat sometime soon.*: George has **internal intentions and desires** that motivate his actions. However, George also has **internal critics** that can prevent him from acting on a desire or goal. Desires and critics can compete with each other to control plan and action selection. He is probably not annoyed enough to actually harm the cat, given **social ideals** and more practical consequences.

This review covers a fraction of the true complexity embedded in this scenario. Ordinary, day-to-day sequence of events like these require most of the unique properties of human cognition that inform and enable our expert

competencies. The above scenario illustrates many observations that come from both psychological and cognitive science investigations into the behavior of human beings.

We can extract from this the following general properties of human decision making:

- People know more than one way to do almost everything
- People are able to pursue multiple goals simultaneously
- People are able to prioritize among competing goals
- People are able to adapt to changing circumstances
- People are sensitive to changes in context
- People are influenced by their current emotional state
- People are able to monitor their own actions and intervene when things go wrong
- People are able to operate reflectively, applying their planning and decision making capabilities to a subset of their own thoughts

5.2 Architectural Overview

The design of the CHIP decision making architecture is show in figure 5.1. This design is motivated in part by the observations in the previous section.

The concept of reflection raises a critical issue, not easily illustrated in the diagram. The entire architecture is operating on two levels simultaneously:

1. **The deliberative layer** in which the goals represent states of the external world that the system is trying to achieve and in which monitoring is applied to actions in the world. Thus, at this level one might have the goal of finding a pen and then select a plan for looking for the pen; later, during execution, this layer would detect a failure in plan execution when the pen isn't where it was expected to be
2. **The reflective layer** in which the goals represent states of the deliberative layer and in which monitoring is applied to one's deliberative thinking. At this level, one might have the goal of finding a plan for finding the pen, select a planning method such as case-based reasoning; during the planning process, this layer would detect a failure if the analogy between the retrieved case and the current situation doesn't yield a solution. This will result in a modification of the deliberative layer over time

The complete architecture is quite complex and so our description of it will proceed in increments, starting with a simple core and then layering on additional elements that lead to the capabilities above.

5.2.1 Biological realism

In the rest of this chapter we describe in some detail the mechanisms involved in building the CHIP reasoning and decision making system. In doing do, we will mainly use terminology drawn from AI and computer science. This has the advantage of providing a relatively detailed computational model; but it comes with the disadvantage that it decouples the description from its biological inspirations. There are a few reasons for this presentation choice. First, human decision making is extremely complicated and description tied more closely to the neurobiology may lose the forest for the trees. Second, less is know about the neurobiology of human decision making than is know about areas "closer to the periphery" such as perception, actuation, memory and low-level learning mechanisms. Third, we have more than ample phenomenological evidence explicates what humans do (as opposed to how they do it) and one of our principal goals is to be faithful to these observations about human behavior.

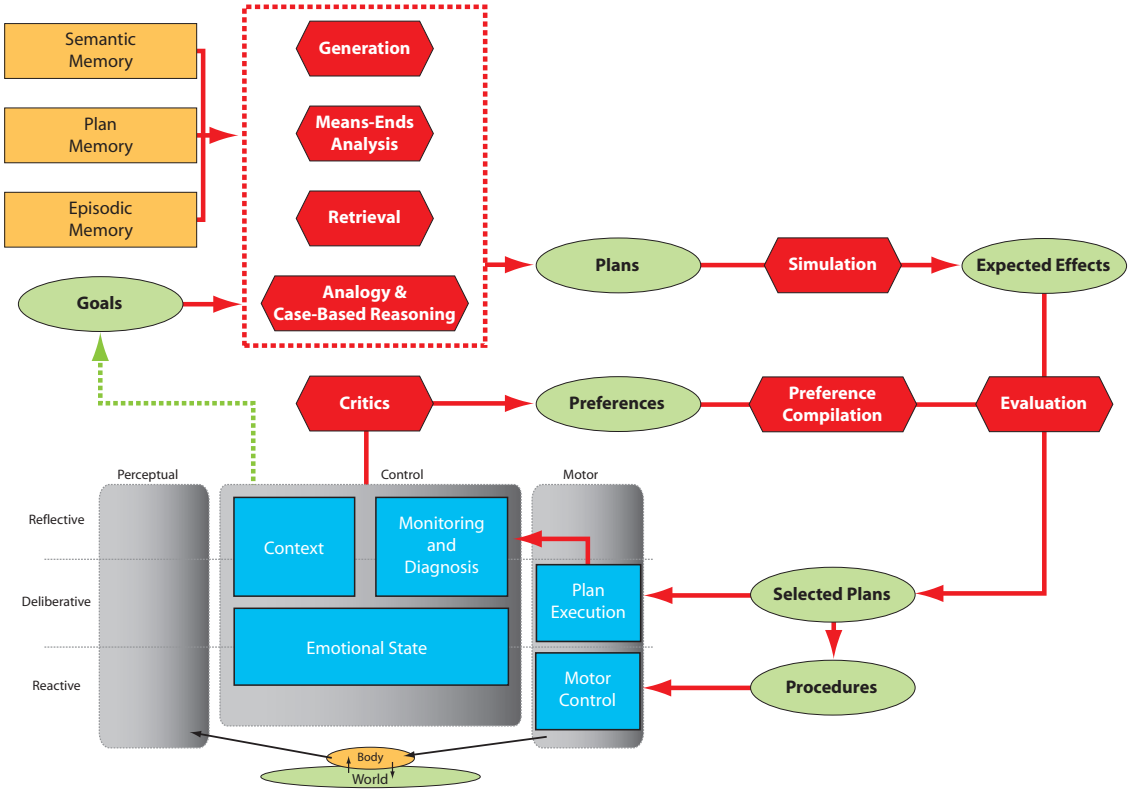


Figure 5.1: The CHIP Decision Making Architecture: Decision making represents a major loop within the CHIP Architecture. This diagram illustrates shows the loop structure flowing from current reactive-layer states through goals, plans and preferences to modify the sensory-motor state of the reactive-layer resulting in another loop traversal. Goals are used to retrieve plans from multiple memories. Plans results in a set of predicted outcomes which are ranked according to an evaluation of current preferences and resources required to result in a set of active plans that alter the behavior of lower level loops.

In particular, it may seem odd that in the rest of this discussion we will use the vocabulary of rational choice and computational decision-theory to describe how human decision-making works. Given that humans are often manifestly not rational this might at variance with our goal of staying faithful to human behavior; however, what decision theory guarantees is approximate rationality with respect to one's personal goals and beliefs. That is to say we can be *locally* rational in a given contextual state while being *globally* irrational relative to other possible states. And in fact this is a natural consequence of the way in which emotions, goal context, task context, environmental context, priming context (or recent history) and reflective state all influence the dynamic behavior of the system.

It might seem rather unlikely to suppose that the human brain computes decision-theoretic quantities in order to make its decisions; nevertheless there is a growing body of neurological evidence, referred to as "Neuroeconomics" that suggests that this is actually the case. We will return to a brief review of this literature after completing the description of our architecture.

5.2.2 Decision-making agent system

The CHIP components that contribute to decision-making may be thought of as a network of mutually reinforcing and inhibiting agents; control ripples through this network as new events occur much in the style described in Minsky's Emotion Machine [45] and Singh's related work [72, 69, 71, 70]. These agents form clusters within which most communication occurs through direct data passing connections. However, in some cases shared information must be preserved for some time or shared among many parties. In these cases, the agents use an area of a three

tiered blackboard [33, 10, 17, 51], introduced in section 1.4. This couples the decision-making and control part of the overall CHIP architecture to its perception and actuation components.

The decision making architecture has the following types of agents:

- Agents that post goals
- Agents that prioritize goals
- Agents that develop plans
- Agents that analyze plans
- Agents that prioritize plans
- Agents that monitor plan execution
- Agents that diagnose breakdowns in plan execution
- Agents that propose fixes to failed plan execution
- Agents that maintain the current context
- Agents that maintain a representation of the current emotional state

5.2.3 Process flow

The deliberative layer of the CHIP decision-making architecture runs continuously. It maintains a priority ranked queue of active goals and will dispatch as many of these as possible, limited only by the availability of resources. For each goal it attempts to find plans capable of achieving it. Each such plan is analyzed to find its resource needs, quality of service and side effects. This information allows the architecture to critique each plan, noticing whether the side effects of the plan are undesirable, whether the quality of service delivered is desirable or irrelevant, and whether the resource costs of the plan are excessive. The critique of the plan is expressed as a set of comparative preferences over these properties of the plan. Given such information, the plans for each goal are prioritized and the most desirable plan is selected.

As the plan is executed, it is monitored to make sure that its actual behavior is consistent with expectations. If not, the breakdown is diagnosed and the diagnosis is then used to select a way of fixing the plan if possible (in effect reinvoking the whole planning process). If no fix is possible, then a different plan is considered.

While these activities are proceeding, the blackboard is constantly updated with new information about the physical context (e.g. changes of time and place), the task context, and emotional state (e.g. something has happened to make the system angry). The agents that prioritize goals and plans are sensitive to all of the above aspects of context; different agents are active in different contextual situations, resulting in different goals and plans being selected.

The reflective layer of the CHIP decision-making architecture runs a similar process; however, while the deliberative layer is concerned with acting on the world, the reflective layer is concerned with managing the deliberative layer. It is this layer that allocates resources (such as time) to the deliberative layer and that selects the methods by which the deliberative layer plans. The reflective layer monitors the deliberative layer's execution and detects breakdowns in its processes. For example, a failure to find any plan for a goal is a breakdown at the deliberative layer which is captured by the reflective layer; in response, the reflective layer might suggest trying a different planning mechanism, or it might suggest abandoning the goal altogether. The reflective layer can also intervene in the deliberative layer's plan selection process, for example, pointing out drawbacks of a particular plan that are only evident from a self-conscious or socially-conscious point of view.

5.3 Planning

The CHIP decision making architecture is a goal-driven, top-down mechanism. CHIP maintains a queue of active goals and then repeatedly tries to find plans capable of realizing those goals.

5.3.1 Plan representation

CHIP represents plans using a very general formalism with roots in the plan calculus of the "Plan Calculus" of the Programmer's Apprentice [59, 66, 58]; it includes a hierarchical nesting of components, each with input and output ports connected by data and control-flow links. Each component is provided with prerequisite and post-conditions. The formalism includes branches and join points. In addition, the CHIP plan formalism includes plan steps whose plan elaboration is deferred until runtime; this allows CHIP to represent an idea such as "I'll drive down the street, see how much traffic there is and then plan how to proceed at that point". A representative plan is shown in figure 5.2

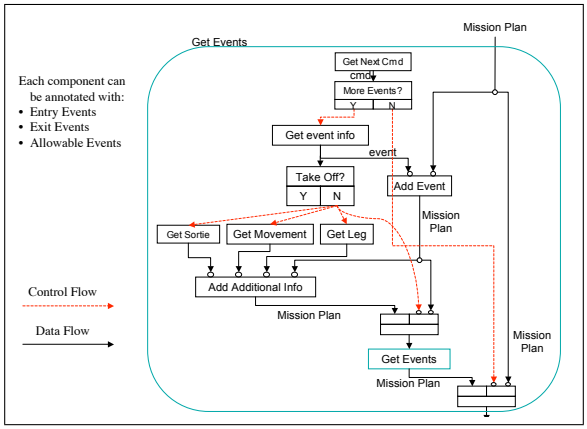


Figure 5.2: An Example Plan: Important features of the "Plan Calculus" are preconditions, actions (subplans), expected postconditions, branches, and join points. Note that this plan component is recursive.

5.3.2 Planning mechanisms

CHIP has several planning mechanisms:

- Hierarchical Task Net planning [19, 20] retrieves an entire plan from "plan memory"; however, the plan is only partially elaborated, containing both specific actions and explicit subgoals. The subgoals are, in turned planned for using any of the CHIP planning mechanisms.

- Means Ends Analysis [18, 50] is driven by considering the most important difference between the goal state and the current state of the world. It uses an "operator difference table" to find actions capable of reducing the difference and then recursively plans both how to get from the initial state to a state in which the preconditions of the operator are satisfied as well as how to get from the state following the application of the operator to the end state.

- State-space generative planning [22] works by chaining together primitive operators either working forward from the initial state or backwards from the goal state.

- Case-based or analogical planning [73] works by retrieving prior experiences from episodic memory and then constructing an analogy between the retrieved experience and the current problem. It then transfers the solution in the prior experience to the current situation.

5.4 Plan analysis and prioritization

Normally the planning process leads to more than one possible plan for each active goal. Each plan is simulated internally in order to make predictions about side effects of the plan, its quality of service and its resource needs. Plan simulation need not be done for retrieved plans, since the results of the simulation are cached with the plan; for plans generated on the fly, the simulation is produced as a by-product of the plan generation process. Plans requiring resources not currently available are put aside (note that by resources we mean both external resources such as tools and internal resources such as the use of perceptual channels or limbs). At this point in the process, a group of agents establish a set of preferences over these properties. Preferences are comparisons between two sets of these properties; for compactness, preferences are stated *Ceteris Paribus* (i.e. everything else being equal).

For example, if one wants to watch a movie, some of the properties of interest might include the video and audio quality and whether the length of delay incurred in obtaining the video. One plan might involve watching the video on your phone or IPod, while another might involve getting a DVD from Netflix. Simulation reveals that the first has low quality but short delay, while the second has high quality but long delay. One plan prioritization agent might state that delay is very undesirable (which is the same as saying that a long delay is much less desirable than a short delay) while another agent might say that video quality is more important than audio quality.

As another example consider our introductory scenario. George is looking for his pen, so he considers various plans he might enact. In one of them, he asks his wife for help; but she'll probably remind him that he's always losing stuff. He regards this as criticism and he prefers not to be criticized. In the other hand, if he asks he'll have company. Apparently, George prefers not being criticized to having company and this is added to his preference set for this goal.

This process of plan evaluation through preference elaboration draws heavily on representations of self and common sense knowledge about the world. This entire body of knowledge is context dependent: George might prefer avoiding criticism to having his wife's help at the moment; but if he's more frustrated, or under greater time pressure, his preferences might well be different.

The set of active preferences is then converted into an evaluation function that is capable of rank ordering the competing plans; this uses techniques based on those in [42, 67, 53]. Technically, this function satisfies the requirements of a decision theoretic utility function and so the entire process can be regarded as an application of decision theory.

Within the context of decision theory the CHIP architecture can include others issues such as the likelihood of each plan succeeding, the value of the resources that are used during the plan, the likelihood and consequences of failure. These are all dealt with by making plan evaluation depend on *Expected Net Utility*: the difference between the expected value of success, the expected cost of failure and the cost of using the resources required by the plan. More detail on this is provided in [67] and [53].

5.5 Goal prioritization

Preferences are used in CHIP to select between different plans for the same goal. However, the same techniques are also used for prioritizing goals. Goal prioritization agents state preferences between different goal conditions. There are then converted into a utility function and this is used to assign a ranking (or value) to the various active goals.

We been describing this process so far as a sequential process, with each step monitored by the reflective layer to catch breakdowns. This is not actually a commitment of the Architecture; it is equally acceptable for an implementation to run many planning techniques in parallel, with the reflective layer selecting between their results. This might, in fact, be more biologically plausible, since neurological computation is slow, but ample parallelism is available.

We defer discussing the algorithm for building the ranking function from the set of preference statements until the the end of this chapter, see section 5.12.1.

5.6 Plan selection

The priority value of each goal is then used to scale the net expected value of its best plan. These best plans are then rank ordered and sequentially assigned their required resources. If the resources required by the next plan are no

longer available because they've been assigned to higher ranked plans, then the goal and plan are abandoned.

5.7 Plan monitoring, diagnosis and repair

During its execution, the plan is monitored to make sure that things are proceeding as expected. If not, the failure is diagnosed and repair agents are invoked to find a patch if possible and if not planning is reinvoked to find another plan.

A plan is not simply an abstract procedure; it also includes a set of constraints about what conditions must be true at a given point in its execution. In particular, it includes statements about the preconditions and post-conditions of each step; in addition, it includes links between the preconditions of each step and the post-conditions of those other steps that establish these preconditions. As the plan is executed, CHIP's perceptual components are tasked to validate these conditions.

Should one of these checks show that the expected condition has not been achieved, then model-based diagnosis [12, 15, 32, 86, 16] is invoked. Driving this is the dependency structure in the plan being executed as well as models of all the resources (both internal and external) being used. Model based diagnosis traces backward from the failed condition attempting to deduce what parts of the plan failed and why. Possible diagnostic explanations include the possibility that a plan step is fallible, that a resource is broken, or that the environment didn't match assumptions.

At this point, repair knowledge is invoked to try to fix the most likely cause of the breakdown. Repair knowledge may be very general purpose knowledge such as that described in [79, 70] or it might be very domain-specific. By default, if no specific repair knowledge is available, then the plan is abandoned and another plan is tried.

The CHIP reflective layer executes the same form of control over the deliberative layer as the deliberative layer executes over base level activity. The reflective layer intervenes when the deliberative layer reaches an impasse. This can happen in a number of situations: The deliberative layer can fail to find a plan using its currently selected planning approach. Possible diagnoses include the absence of stored plans (e.g. if the current planning approach is hierarchical task network planning) and a possible repair is to use a different planning technique. Another type of impasse involves a breakdown in the preference exploration process; inconsistent preferences lead to a circular preference graph enabling easy diagnosis. A possible repair is to remove one of the preferences in the cycle.

The reflective layer also has the option of creating a new subgoal that re-tasks the planning machinery to derive a new repair plan for a given failure by deliberating from the original plan, the nature of the failure and episodic memories of past repair experiences.

All of these processes combine to make a system based on the CHIP architecture highly adaptive. It can respond to unexpected circumstance in the environment and it can adaptively learn how to recover from common classes of failure.

5.8 Contextualization

Goal posting, plan generation and planning agents are all contextually sensitive; each of these agents is active in only some contexts and inactive in others. In some contexts, the effects of one agent may be overridden by the actions of another agent deemed more salient for that context.

The fact that all the agents involved in decision making are contextually sensitive leads to adaptive behavior. A goal that seems important at one time will seem irrelevant at another. A plan that seems particularly useful in one place, will be less so in another. In addition, dynamic changes of the context will necessarily result in a cascade of mental state revisions in which the relative priority of goals and the relative attractiveness plans changes. This in turn can cause a currently executing plan to be abandoned on the grounds that some other goal has risen to greater priority.

Context is a multi-dimensional notion including location, task, time of day, social setting and others. This contextual sensitivity accounts for the fact that playing music through loudspeakers is a reasonable idea when at home alone, but not when you're in your office (location context), or that using a phone makes sense in your office but not when driving (task context), or that drinking coffee is OK in the morning but not near bed-time (temporal context).

Emotions play an particularly important role in this process. Just as various aspects of context affect which goals, plans and preferences are active, so too does emotional state. This means that radically different behaviors can be produced in different emotional states. In our opening scenario, George might think about killing the cat, but he would never go so far as to actually consider that as a real goal in his normal emotional state, much less consider which plans for killing the cat best satisfy his preferences. But we all have said things, when angry enough, that we wouldn't have otherwise said or failed to do tasks we normally would attend to when sad or depressed enough.

Finally, emotional states can be triggered (or suppressed) in several different ways. At the reactive level, basic emotions such as fear are triggered by raw perceptions; many people have fear reactions to visual stimuli even vaguely suggestive of a snake and with reaction times faster than is usually associated with object recognition. But many emotional states, are triggered in a more deliberative way; George's frustration at not being able to find his pen is perhaps an example. Finally, at the reflective level, one might self-consciously argue oneself out of going into a particular emotional state on the grounds that it's socially unacceptable. However triggered, emotions serve to enable and disable different critics, goal posting agents, and plan proposing mechanisms, leading to different ways of thinking and acting.

Each CHIP decision making agent has a number of contextual preconditions that state under which conditions its relative. Each contextual dimension of these preconditions are situated in an ontology. For example, for spatial context, we distinguish outdoors from indoors, offices from public spaces, etc. The preconditions of any agent will fall at the most general relevant position in the ontology. Given the description of the current context, more than one agent might be relevant and these will all fire adding their assessment of preferences to the decision making blackboard. However, if one agent's conditions are strictly more specific than those of another, it will block the other agent's activation.

5.9 Learning

The CHIP architecture naturally supports several forms of learning that take place within the decision-making subsystem. Some of these, such as chunking [38] or the formation of macro-operators [23] and explanation-based learning [46] have been employed in other cognitive architectures. These learning mechanisms all arise as a side-effect of CHIP's planning methods; macro-operators (or chunks) are formed and cached as generative planning techniques produce useful sub-plans. Explanation-based learning is guided by the production of the causal structure of a plan during decision making.

However, there are other types of learning that take place within the CHIP architecture that are facilitate by its unique features. One of these is the elucidation of the structure of the prioritization agents that rank goals and plans through the explication of preferences. "Preference learning" consists of learning the association between context and particular agents. This type of learning takes place in a self-supervised fashion, using explanation-based techniques. If a particular choice of goal or plan results in unwanted results, the analysis of the dependency chain leading to that choice can deduce that a particular agent should not have been active in that decision making context. This then results in a change to the contextual preconditions of the agent. This form of learning can only occur in an architecture, such as CHIP, that explicitly represents context and preferences.

Another unique form of learning that can occur in the CHIP architecture arises from the presence of a reflective layer that observes and intervenes in the behavior of the deliberative layer. Just as the deliberative layer observes and acts on the world, the reflective layer observes and acts on the deliberative layer. Thus, learning mechanisms that operate in the deliberative layer can also operate in the reflective. However, these higher-order learning mechanisms result in improved performance of the deliberative layer. In more common sense terms, learning at the reflective layer is learning *how to think better*.

Examples of learning that is enable by monitoring the internal representations of the decision-making system itself include:

- Monitoring the success and failure rates of different plans and updating the expected failure rate of those plans
- Learning when specific types of analogies apply

- Learning better indexing techniques for episodic and plan memory

It is important to notice that these forms of learning are not the result of unique learning mechanisms. In fact, they system uses many known learning techniques. Rather, the uniqueness emerges from the interaction of the standard techniques and unique representations; in the case of reflective learning this involves the representation of internal mental states.

5.10 How the CHIP decision making architecture achieves its goals

In section 5.1.1 we made several observations about human behavior. Here we briefly review these observations and show how similar properties emerge from the CHIP architecture.

- **People know more than one way to do almost everything**. In the CHIP architecture, every goal is reduced to a plan. The CHIP architecture provides a "plan memory" which acts as a library for already known plans. In addition, it provides for several different planning mechanisms that allow it to use its prior experience its knowledge of existing plans and its knowledge of the properties of primitive operators in order to generate new plans that are added to the library.

- **People are able to pursue multiple goals simultaneously**. In the CHIP architecture there is an agenda of active goals. The plan dispatching part of the architecture will dispatch plans for as many goals as possible, limited only by availability of resources with which to execute the plans.

- **People are able to prioritize among competing goals**. In the CHIP architecture, a network of agents rank orders goals by reasoning about which properties are more desirable in the current context. Similarly, this network of agents ranks the various possible plans for each goal. Decision-theoretic techniques are then used to select the most attractive goals and plans.

- **People are able to adapt to changing circumstances**. In the CHIP architecture, all of the agents that analyze and rank plans and goals are sensitive to context and to the availability of resources to implement these plans. If circumstances change, this is reflected in a change of CHIP's context model; this propagates through the agent network, changing the ranking of goals and plans. If resources change their availability status or if conditions in the world change such that the preconditions of previously attractive plan are no longer in effect then this is reflected in a changed assessment of whether the plan is executable. All of these allow adaptation to change.

- **People are sensitive to changes in context**. In the CHIP architecture, all agents involved in prioritizing and selecting goals and plans are sensitive to the current context. A change of context results in a new configuration of these agents which in turn results in changed assessments of which goals have higher priority and which plans seem more attractive.

- **People are influenced by their current emotional state**. Just as with other aspects of context, all agents involved in decision making are sensitive to the emotional state. Changes in emotional state, change which agents are active, leading to changed assessments of which goals and plans are desirable.

- **People are able to monitor their own actions and intervene when things go wrong**. In the CHIP architecture, all plans are monitored during execution to make sure that expected preconditions and postconditions of the plan steps actually obtain. These checks establish top-down focus for the perceptual system, causing observations to be made that confirm (or deny) that the expected conditions obtain. If the expected conditions fail to hold, model-based diagnosis is invoked to characterize the breakdown. Repair agents are then invoked to attempt local corrections; if there are no available local patches, then re-planning is invoked to find another plan for the goal. Thus, a system based on the CHIP architecture will be able to notice and repair breakdowns.

- **People are able to operate reflectively, applying their planning and decision making capabilities to their own thinking**. The CHIP architecture operates at three levels. The highest of these, the reflective layer, bears the same relationship to the deliberative layer as the deliberative layer bears to the world. This means, that a CHIP based system can decide how to pursue a problem, notice that it isn't making progress in reasoning about that problem, figure out why and then decide to try some other approach to thinking about the problem. Finally, it can learn from all these experiences, allowing it to learn how to think more effectively.

5.11 Biological grounding

Decision-theoretic plan evaluation lies at the core of our decision making architecture; this evaluation is with respect to a set of context and emotionally conditioned preferences that result from an activation process rippling through a network of decision-making agents as described earlier. Thus, although the CHIP decision making architecture can be seen as performing a form of rational choice, this choice is respect to a set of changing, ideosyncratic, and possibly inconsistent beliefs and desires

Thus, expected utility is a critical decision variable which, if our approach is biologically ground, should have a neurological correlate. Over the last several years, there has been a growing body of evidence that this is true. In the rest of this section we will describe several results from the field of Neuroeconomics [27, 28, 29, 30, 78, 88] that provide suggestive neurological evidence for the conclusion that the brain does compute expected utilities that guides its decision making.

One of the earliest such experiments is due to Platt and Glimcher [55] who trained rhesus monkeys to participate in repeated rounds of a simple lottery while the activity of nerve cells in the posterior parietal cortex was monitored. The monkeys were required to visually oriented towards a visual signal (a light) either on their left or right and received a fruit juice reward if they oriented correctly. Platt and Glimcher systematically varied either the relative probabilities that the left or right lights would be selected at the end of each round or the size of the reward associated with each. It should be noted that expected utility is the product of the probability and the size of the reward. Platt and Glimcher found that some signals in certain parietal neurons correlated very strongly with expected utility.

Glimscher and Rustichini [30] survey several other such experiments: For example, an fMRI study by Knutson and colleagues that shows that activity in the human striatum is correlated with the magnitude of the monetary reward subjects earn during lotteries; another study by Paulus and colleagues have shown a similar result in the human posterior parietal cortex. Yet another study by Breiter and colleagues found that the activity of the sublenticular extended amygdala encoded the desirability of each of several different lottery arrangements.

In establishing the biological plausability of the decision making architecture it is instructive to review the important representational components that are not treated elsewhere in this document. Conceptually, plans are simply sequences of anticipated mental states, including gating conditional stages that enable preconditions, postconditions and branching. As a plan is executed, a decision point can perform AND gating with signals from the perceptual system to evaluate pre-conditions and post-conditions. A failure of the gating function informs the reflective system. Branch points are identical except there the next stage is gated by the excitation of an associated region of the perceptual system representing the branching condition.

The plan itself can be stored as associations between cortices and mediated by the hippocampus. Mentally replaying a plan (or episode) sequences the appropriate cortical systems through the partial states associated with the plan. Thus execution and deliberative review of a plan are similar except that perceptual expectations are used instead of perceptual state.

While plans may be computationally described as discrete data structures, the simple components of these data structures can independantly map to complex cortical states and there is strong evidence for eachstates (hippocampal associations), conditions and branch points (inhibiting and amplifing effect of gating in the Basal Ganglia) and join points (thresholds in a set of neurons). The key frame, or triggering association for a plan, can become a marker for the sequence allowing it to be treated as a discrete chunk.

5.12 Summary

The CHIP decision-making architecture is a multi-tiered, adaptive problem solving system based on preference driven, decision-theoretic techniques. All of the agents contributing to decision-making are sensitive to context and emotional state, meaning that changes in context, or changes in the system's emotional state, lead to changes in the relative importance of goals and plans. Plan execution is accomplished both deliberatively and procedurally through interaction with the sensory-motor systems. Plans assert behaviors into the actuation system that are monitored through interaction with the perceptual system. Breakdowns are diagnosed and the diagnosis is used to select repair strategies.

This overall structure leads to the following properties of the CHIP architecture:

- It can find more than one way to do almost everything
- It is able to pursue multiple goals simultaneously
- It is able to prioritize among competing goals
- It is able to adapt to changing circumstances
- It is sensitive to changes in context
- It is influenced by its current emotional state
- It is able to monitor its own actions and intervene when things go wrong
- It is able to operate reflectively, applying its planning and decision making capabilities to its own thinking

5.12.1 Appendix: an algorithm for preference evaluation

CHIP uses a graph theoretic to convert a set of preferences into an utility function. Each preference statement consists of two conjunctions of parameter-value pairs and a weight stating how strongly the preference is held. Each of these is understood to hold "everything else being equal". The steps in the algorithm are:

1. For each preference statement, find all parameters not mentioned and then generate all possible combinations of values for these parameters. For each such combination, produce a new preference with that combination added to both sides of the preference statement. (In other words since the preference holds everything else being equal, create a new preference making explicit those other circumstances).
2. For each preference statement add the negation of each side to the other side. (Another words make explicit that "Not being criticized" is better than "having company" is really saying that "Not being criticized and not having company" is better than "being criticized and having company". If necessary, perform boolean simplification on the resulting statements so that each simplified statement is a comparison between two conjunctions.
3. Create a graph where each node represents a unique combination of the parameters and their values.
4. Add an arc for each preference statement from the node corresponding to the left side of each preference to the node corresponding to the right side. The weight of this arc is that of the preference statement.
5. For each node examine all of its outgoing arcs. If there are none assign the node value 1. Otherwise, assign the node the maximum value of the product of the arc's weight and the value of the node pointed at by the arc. This can be done using a simple dynamic programming algorithm.
6. The utility function simply looks up the node corresponding to a combination of parameter-value pairs and returns that node's value.

Chapter 6

Language and Representation

6.1 The Capability

We believe that language is much more than a medium of communication. Mounting evidence suggests that language is the great differentiator—the capability that separates the intelligence of the human species from that of other primates, other mammals, and every other animal. Some argue that apes or parrots can be taught some language; some argue they cannot, that they are just engaging in a limited form of stimulus-reponse behavior. The debate is, as they say, academic, because one fact is clear: no other animal has anything like the human ability to use language to tie concepts together and to do it without limit, where *without limit* distinguishes the human level from that of all other claimed ability of animals to understand and use language.

The ability to tie concepts together—an ability intimately connected with language—seems to lie at the the core of our unique ability to think so effectively. This **combinator** capability serves to construct layered, heirarchical symbolic descriptions from diverse, perceptually grounded elements. As we argue in this section, those symbolic descriptions greatly facilitate our ability to accumulate raw experience, real and surrogate, and put that experience to use in new situations. Thus, we see external language as the external manifestation of internal descriptions, cast in representations that mirror what is important to us as we interact with the world throughout our development.

It follows that **any system fulfilling the Biologically Inspired Cognitive Architecture vision must be a system touched by a deep understanding of language, combinators, and representations and what they jointly contribute to intelligence.**

Because language is deeply intertwined with our combinator capabilities, it is as important for the organization of internal thought as it is for the communication of thought externally. Saying things to ourselves may be even more important than saying things to others. Hence the following hypothesis: Language has evolved so as to enable us to express those concepts that are most useful to capture in our internal representations.

Of course, the language faculty cannot be studied in isolation. Quite the contrary; we believe that excessive concentration on purely symbolic thought has had a retarding effect on progress toward understanding intelligence. Our language system is essential not only because of what it can do but also because of what it can marshall. Any child, if told "John kissed Mary" and asked "Did John touch Mary" everyone says "yes" and reports that that they know it must be true because they imagine the scene and read the contact off of the imagined scenario as if it were real. Any adult, if told "Watch out, there an angry grizzley just ahead" will look around for a tree to climb, not because of any built-in rule-based system that reasons about bears, but because he imagines scenarios in which he does or does not climb something. Thus, language has an essential role in stimulating the reuse of sensory-motor experience to deal with imagined worlds, and we conclude that **any system fulfilling the Biologically Inspired Cognitive Architecture vision must include an inquiry into how the language system is connected in tight loops with perceptual systems to solve problems through imagination.**

6.1.1 Biological grounding

The ability to combine concepts without limit seemed to emerge rapidly about 50,000 years ago, probably coevolving rapidly with the language faculty. Before that, although homo sapiens, of a sort, were already in place for 100,000 years or so, we did not amount to much, leaving few artifacts behind. After that, we emerged from Africa, spread

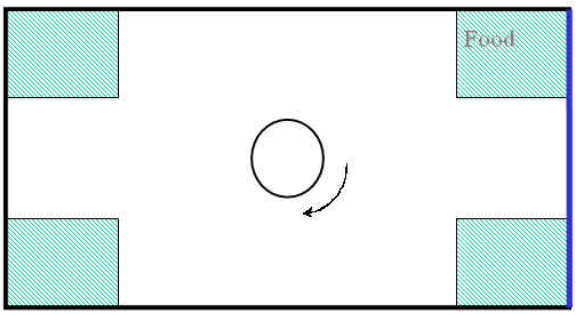


Figure 6.1: In a series of experiments, a rat, a child, or an adult watch while an experimenter hides food or some other object in one of the corners of a rectangular room. After disorientation, only the adult can use the blue wall to determine exactly in which corner the object was placed. Jamming the adult's language faculty reduces the adult to the level of a rat and child.

throughout the world, killed off a great number of species that provided convenient food or occupied similar niches, and in the blink of an evolutionary eye, painted the caves at Lascaux in France and got to work on the pryamids in Egypt.

The dispersion of humans over the world must post-date the evolution of language, since there is no detectable difference in basic language capacity among contemporary humans. Tatersall takes language to be "virtually synonymous with symbolic thought," implying that externalization is a secondary phenomenon (see Ian Tattersall, Becoming Human, 1998). Jared Diamond calls the emergence of language the "great leap forward", and ascribes it to the result of some genetic event that rewired the brain, allowing for the origin of modern language with a rich syntax that provides modes of expression of thought, a prerequisite for social development and the sharp changes of behavior that are revealed in the archaeological record, and presumably occasioned the trek from Africa.

Rat and developmental psychology together provide intriguing modern evidence for the role of language and combinators as a key distinguisher of human thinking. Imagine a rectangular room, with walls painted white. Each corner has a basket or other place that can hide food or a toy. Place a rat, small child, or human adult in the center, and have that subject watch you hide the food or toy in one of the four baskets. Then, spin the subject around to disorient it, and watch where he goes to get the hidden prize.

All three go with equal probability to the two diagonally opposite corners where the prize can be, as they should in a rectangular room; all three lack interest in the two corners where it cannot be. Rats are pretty smart.

But now, paint one of the end walls blue, so as to break the symmetry. Rats, which have perfectly good color vision, still go to diagonally opposite corners. So do small children. Only human adults get it right. Children only become adults at about five, and after an elaborate series of experiments, the transition is correlated only with the onset of the words *left* and *right* in the phrases the child uses to describe the world. The words are manifest evidence of a new representation that the child has formed that uses combinators to combine information from both geometric and color subsystems (the left corner of the narrow, blue wall versus The left corner of the narrow wall).

Next, in experiments conducted by Elizabeth Spelke and Linda Hermer-Vazquez [1999], a human adult is asked to repeat a story back to a reader as it is read, as if doing a simultaneous translation, but from English to English instead of to another language. Amazingly, this reduces the subject to the level of a rat, breaking the subject's ability to use the blue wall to advantage. It seems that engaging the language faculty with the say-back task blocks the ability of the language subsystem to combine information about shape with information about color, which are known to be handled in different parts of our brain.

All this is highly informative for those interested in a computational account of human intelligence. Cast in language familiar to AI researchers, once you have combinators you can construct semantic nets. Once you have semantic nets, you can build frames, and on top of frames we can construct all sorts of representational structures, and with descriptions expressed using appropriate representations and linked together at higher levels with other

combinators, such as *cause*, you have stories, and when you can match stories together, with a controlled level of type difference, you have analogy and you are ready for abstract thinking. So, Spelke and Hermer-Vazquez may well have established a tight link between the key ability to combine concepts without limit and the human language faculty, and they have shown that without access to the combinators, we think with both boots off.

6.1.2 Language as Evidence for Representations

We presume that something easy to express in language must come readily out of or fall readily into an internal representation, therefore the structure of language provides rich clues about the structure of those internal representations. When we say "a person is an animal" we acknowledge the existance of a representation for class; when we say "the person walked along the street toward a cafe" we acknowledge the existance of a representation for trajectory; and when we say "a person's speed increased" we acknowledge the existence of a representation for transition. We conclude that **any system fulfilling the Biologically Inspired Cognitive Architecture vision must exploit and account for the representational clues offered by language.**

We are obsessively interested in representation because we believe that representations are the tools with which humans build models that serve to explain the past and predict the future. For example, class, trajectory, and transition are among the representations that make it possible to understand the world around us and accumulate episodic memories. Those memories make it possible to discover regularities, which in turn make it possible to be surprised only when we should be, rather than with each new experience. All of these faculties, in turn, support the acquistion of abstract representations. Buying and selling, for example, can be viewed as the movement of the thing bought or sold in an abstract posession space built on our understanding of movement in physical space.

To understand how systems of neurons generate human-like thought, we believe it is essential to develop an understanding of the representations a particular neural subsystem comes to support. Otherwise, it seems unlikely that we would be able to explain why the genetic distance between primates and humans is small, yet the capability difference is vast, and without that understanding, it is unlikely that a neurologically-inspired system could be made to work at a human level.

Fortunately, many diverse, higher-level representations of the world have emerged during the first 50 years of research in Artificial Intelligence and allied fields. Unfortunately, it is hard to identify systems that exhibit more than one or two representations with first-class status. It follows that it is hard to identify systems that support research along all the diverse dimensions that make human concepts what they are.

We conclude that systems fulfilling the Biologically Inspired Cognitive Architecture vision should include steps such as the following, aimed at establishing a cognitively complete set of representations closely aligned with natural language:

- Catalog existing representations
- Identify and fill prominent gaps
- Build a system incorporating a dozen or so of the most important representations
- Develop mechanisms that observe language streams and produce descriptions in appropriate representations
- Accumulate experience by blending descriptions produced in diverse representations.
- Build multidimensional links by observing coincidences emerging from representational specialists

By accumulating representation-specific memories, appropriately blended together and generalized, it becomes possible to accumulate a great deal of representation-specific experience. For example, Kevin Stolt (2006), in his undergraduate thesis completed in May, 2006, reported on an implemented system that used a stream of trajectory-describing sentences to populate a symbolic self-organizing map. The experience captured in the map enabled Stolt's system to answer fill-in-the-blank questions, inserting, for example, *bird* in "The xxx flew to the top of a tree," and *flew* in "The bird yyy to the top of a tree."

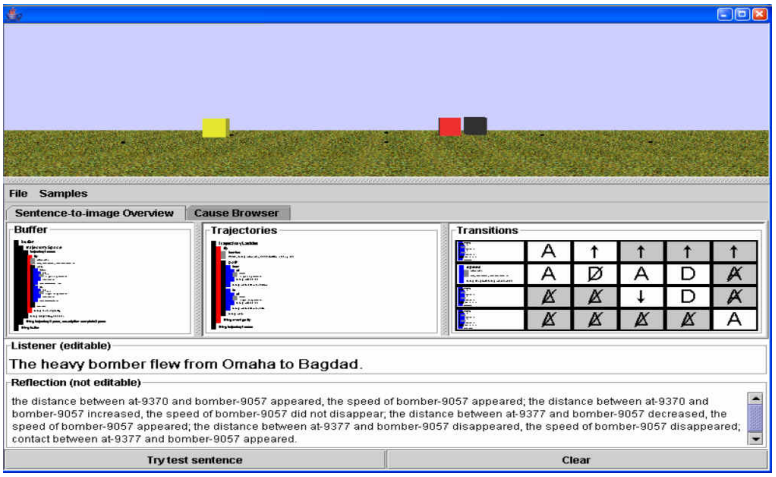


Figure 6.2: Jackendoff's trajectory representation and Borchardt's transition, at work in a visualization system that tries, as best it can, to visualize the motion of bombers.

Once first-tier experts start building memories intramurally, within a single representation, we conceive that another tier should similarly observe and record regularities intermurally, just as humans do, among multiple representations. Equipped with such memories, a system would be able to respond to questions expressed in one representation, such as "The the bird flew to a tree, what happened?" (path in a trajectory representation) with answers expressed in another "The bird came into contact with the tree" (transition representation).

The first step toward building a representationally rich memory system for a Biologically Inspired Cognitive Architecture is to catalog the well-known representations, especially those developed by linguists, such as Ray Jackendoff's trajectory representation. Representative examples of well-known representations include the following[1].

Identifier	**Domain**	**Origin**
Threads	Class relationships	Greenblatt and Vaina (1979)
Trajectories	Objects moving on a path	Jackendoff (1983)
Transitions	Changes in value	Borschardt (1994)
Force dynamics	Causal relations	Talmy
Conceptual dependency	Primitive acts	Schank and students (1982)
Time intervals	Time intervals	Allen (1984)
Cause	Causal relations	Rieger (1976)
Qualitative physics	Object interaction	Forbus and many others (1988)
Somatic Markers	Emotions	Damasio (1994)

The relative utility of the representations may lie along a Zipf-like curve; that is, once we have all the representations we need, and rank them according to frequency of use, it may be that utility emerges as inversely proportional to rank. Unfortunately, not everything that we need to represent can be captured in an existing representation. Conceptual dependency, for example, is weak on activities you can perform with your body (you can eat, but you cannot dig or point). Likewise, existing representations for trajectory are weak on path blockers and containers (You can travel along a path, but nothing can prevent or interfere with your movement). We have maps but nothing particular for viewer-centric spatial arrangement or for visibility (who can see what). Accordingly, any Biologically Inspired Cognitive Architecture needs to answer the following questions;

- What high-frequency representations are missing?
- What representations are found in what stages of development?

[1]Many of these examples are based on earlier work, and many have been complemented or superceded by more recent work.

- What representations constitute a basis set for constructing others?
- How much coverage can we get out of the top dozen representations?

On a still more ambitious level, we note that memories are not just of individual class relations, trajectories, transitions, or quanta of meaning, or even pairs tied together with causal relations. Instead, the quanta combine into groups and sequences that tell the stories that enable our analogical reasoning. All children in the Western world learn about Cinderella, and when they are old enough, forever after understand what it means for a particular alignment of circumstances to be a Cinderella story. Later on, the same sort of reasoning by precident emerges as we are educated in law, medicine, archetecture, and military science. All students in military academies, for example, learn about Austerlitz and look for opportunities to attack the center (which may explain, in part, Lee's disaster at Gettysburg).

All this happens even though the analogous entities are of different types. The role of Cinderella can be played by a male or a corporation. The opponent need not be Austrians allied with Russians.

All this can happen because we are able to combine reprentational quanta, without limit, into stories. And fortunately, we do not have to live a story to experience it; we can have the experience in surrogate form, via natural language, providing us with an inexpressible power to benefit from others across time and space. It follows that **any system fulfilling the Biologically Inspired Cognitive Architecture vision must include an inquiry into how it is possible to store and make analogical use of stories expressed in natural language.**

Again, useful representations are in place, including the following, but much remains to be done.

Scripts	Event sequences	Schank and students (1982)
Plot units	Intermediate-level story summaries	Lehnert (1981)
Decision rational	Arguments	Toulman, Lee, Shahdadi (2003)

6.1.3 Language as Communication

Any system lacking an ability to communicate with humans in a natural language such as English is likely to be narrow and inflexible. We bristle at dealing with such systems through button clicks and the like, wishing that we could just tell them what to do and wishing we could ask them questions about how they did it. No matter how sophisticated, systems lacking language communication do not seem intelligent. Thus, we are driven to understand how language has come to be such a good vehicle for expressing observations, questions, suggestions, and commands, and we believe that **any system meant to demonstrate the power of a Biologically Inspired Cognitive Architecture must be one that understands what we tell it and commuicates its observations, answers, and suggestions in natural language.**

Readily available statistical parsers claim to parse everything, even jabberwocky, in the sense that when given a string of words they produce some sort of tree. From the perspective of information retrieval, they are useful, albeit often wrong. From the perspective of Biologically Inspired Cognitive Architecture, they hold limited interest for several reasons:

- Statistical parsers give us parse trees and what we want is instantiated representation.
- Unlike humans, they produce nonsensical results on sentences with structures that lie outside of the training corpus,
- Unlike humans, they require millions of training examples.

For Biologically Inspired Cognitive Architecture, what is needed is a cognitively motivated approach to natural language aimed at understanding how language makes it possible to think. Thus, language systems that address some aspects of meaning constitute a better departure point. One such system, the START system produced by Boris Katz and his many students and collegues, takes a step toward translating English into meaning because it translates

English into a kind of semantic net, consisting of sets of nested ternary expressions, each expression having the form constituent–relation–constituent. Also, aspects of START are cognitively motivated in that the state of its internal base of knowledge affects its interpretation of incoming natural language questions and statements. That is, START's attribution of meaning to natural language utterances is knowledge-driven or knowledge-directed. This is similar to human language processing behavior as revealed in numerous experiments, including the well-known experiments on priming effects.

Other practical considerations drove the START system toward the use of a transition representation lying above the basic ternary relations, so as to enable a cognitively-motivated representation of the temporal unfolding of events. The transition space representation enables START to capture information about what happens during particular types of events and to detect instances of corroboration and conflict between reported events.

The CHIP architecture approach to natural language borrows much from systems like START, but represents a much broader attack on the underlying representations of meaning that language references and manipulates. The entire implementation of the CHIP architecture is involved in processing language: vision or audio subsystems identify words, sequence learning subsystems in the deliberative layer of the perceptual system identify phrase structures according to sequences of subcategories of verbs and nouns, the perceptual system also primes the semantic associations of the constitutent words and together these index into episodic memory and, depending on goal context, may result in subgoals and plans that in turn loop back to into the sensory motor systems. Surprises and inconsistences in this process result in activity in the reflective layer. This may result in a new sensory-motor loop to reread the text, a plan for a sequence of motor routines to request clarification, or an internal, mental replay of recently spoken or read words where some aspect of the original processing is modified to create an alternate interpretation.

Bibliography

[1] D. A. Baldwin, J. A. Baird, M. M. Saylor, and M. A. Clark. Infants parse dynamic action. *Child Development*, 72:708–717, 2001.

[2] J.A. Bednar, Y. Choe, J. De Paula, R. Miikkulainen, J. Provost, and T. Tversky. Modeling cortical maps with topographica. *Neurocomputing*, 2004.

[3] Jr. B.G. Galef. Imitation in animals: history, definition, and interpretation of data from the psychological laboratory. *Social Learning: Psychological and Biological Perspectives, T.R. Zentall and B.G. Galef, Jr. (eds.)*, 1988.

[4] I. Biederman. Visual object recognition. In M. Kosslyn and D.N. Osherson, editors, *An Invitation to Cognitive Science: Visual Cognition (2nd edition)*, volume 2, pages 121–165. Visual Cognition M, An Invitation to Cognitive Science, 2nd edition, 1995.

[5] Jean Bullier. Integrated model of visual processing. *Brain Research Reviews*, 36:96 – 107, October 2001.

[6] G.A. Calvert, C. Spence, and B.E. Stein. *The Handbook of Multisensory Processes*. MIT Press, 2004.

[7] M.H. Coen. Cross-modal clustering. In *In Proceedings of the Twentieth National Conference on Artificial Intelligence. (AAAI-05)*. AAAI Press, 2005.

[8] M.H. Coen. *Multimodal Dynamics: Self-Supervised Learning in Perceptual and Motor Systems*. PhD thesis, Massachusetts Institute of Technology, 2006.

[9] M.H. Coen. Self-supervised acquisition of vowels in american english. In *In Proceedings of the Twenty-First National Conference on Artificial Intelligence. (AAAI-06)*. AAAI Press, 2006.

[10] Daniel D. Corkill, Kevin Q. Gallagher, and Kelly E. Murray. Gbb: A generic blackboard development system. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1008–1014. AAAI, August 1986.

[11] K. Dautenhahn and C.L. Nehaniv (eds.). *Imitation in Animals and Artifacts.* MIT Press, 2002.

[12] Randall Davis and Howard Shrobe. Diagnosis based on structure and function. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 137–142. AAAI, 1982.

[13] S. Dehaene and J. Changeux. A hierarchical neuronal network for planning behavior. *Proceedings of the National Academy of Science*, 94:13293–13298, 1997.

[14] S. Dehaene, E. Spelke, P. Pinel, R. Stanescu, and S. Tsivkin. Sources of mathematical thinking: Behavioral and brain-imaging evidence. *Science*, 284, May 1999.

[15] Johan deKleer and Brian Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

[16] Johan deKleer and Brian Williams. Diagnosis with behavior modes. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1989.

[17] Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy. The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2):213–253, June 1980.

[18] G.W. Ernst and A. Newell. *GPS: a case study in generality and problem solving*. Academic Press, 1969.

[19] K. Erol, J. Hendler, and D. S. Nau. Htn planning: Complexity and expressivity. In *Proc. of the 12th National Conf. on Artificial Intelligence (AAAI–1994)*, volume 2, pages 1123–1128. AAAI, AAAI Press, July 31 – August 4 1994.

[20] K. Erol, J. Hendler, and D. S. Nau. Umcp: A sound and complete procedure for hierarchical task–network planning. In K. J. Hammond, editor, *Proc. of the 2nd Int. Conf. on Artificial Intelligence Planning Systems (AIPS–94)*, pages 249–254, June 1994.

[21] M. Fee, A.A. Kozhevnikov, and R. Hahnloser. Neural mechanisms of vocal sequence generation in the songbird. *Annals of the New York Academy of Science*, 1016, June 2004.

[22] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[23] R.E. Fikes, P. Hart, and N.J. Nilsson. Learning and executing generalized robot plans. *Articial Intelligence*, 3(4):251–288, 1972.

[24] N. Fujii and A. M. Graybiel. Representation of action sequence boundaries by macaque prefrontal cortical neurons. *Science*, 301:1246–1249, 2003.

[25] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall, New York, 2nd edition, 2003.

[26] S. Gilman and et al. D. Carr. Kinematic effects of deafferentation and cerebellar ablation. *Brain*, 99:311–330, 1976.

[27] Paul W. Glimcher. *Decisions, Uncertainty, and the Brain The Science of Neuroeconomics*. MIT Press, October 2004.

[28] Paul W. Glimcher, Michael C. Dorris, and Hannah M. Bayer. Physiological utility theory and the neuroeconomics of choice. *Games and Economic Behavior*, 52(2):213–256, August 2005.

[29] P.W. Glimcher. Decisions, decisions, decisions: Choosing a neurobiological theory of choice. *Neuron*, 36:323–332, 2002.

[30] P.W. Glimcher and A. Rustichini. Neuroeconomics: The concilience of brain and decision. *Science*, 306:447–452, 2004.

[31] D. R. Godden and A. D. Baddeley. Context-dependent memory in two natural environments: On land and under water. *British Journal of Psychology*, 66:325–331, 1975.

[32] Walter Hamscher and Randall Davis. Model-based reasoning: Troubleshooting. In Howard Shrobe, editor, *Exploring Artificial Intelligence*, pages 297–346. AAAI, 1988.

[33] Barbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, July 1985 1985.

[34] S. Jo and S. G. Massaquoi. A model of cerebrocerebllo-spinomuscular interaction in the sagittal control of human walking. *Biol. Cybern*, page accepted for publication, 2006.

[35] S. Jo and SG Massaquoi. A model of cerebellum stabilized and scheduled hybrid long-loop control of human balance. *Biol Cybern*, 91:188–202, 2004.

[36] E. Koechlin and T. Jubault. Broca's area and the hierarchical organization of human behavior. *Neuron*, 50:963–974, 2006.

[37] L. Kovotsky and R. Baillargeon. The development of calibration-based reasoning about collision events in young infants. *Cognition*, 67:311–351, 1998.

[38] J. Laird, A. Newell, and P. Rosenbloom. Chunking in soar: The anatomy of a general learning mechanism. *Machine Learning*, 1:11–46, 1986.

[39] S. G. Massaquoi. A ripid model of cerebro-cerebellar interaction in the stabilization of long-loop arm control ii: two-joint control. *Biol. Cybern*, page submitted, 2006.

[40] S. G. Massaquoi and Z. H. Mao. A multi-input multi-output adaptive switching model of frontocortical and basal ganglionic function in procedural learning and execution. *Neural Netw.*, page in preparation, 2006.

[41] S.G. Massaquoi. A ripid model of cerebro-cerebellar interaction in the stabilization of long-loop arm control i: single-joint control. *Biol Cybern)*, page submitted, 2006.

[42] Michael McGeachie and Jon Doyle. Utility functions for ceteris paribus preferences. *Computational Intelligence*, 20(2):158–217, May 2004.

[43] D. L. Medin, J. D. Coley, G. Storms, and B. Hayes. A relevance theory of induction. *Psychonomic Bulletin and Review*, 10:517–532, 2005.

[44] G. A. Miller, E. Gallanter, and K. H. Pribram. *Plans and the structure of behavior*. Henry Holt and company, 1960.

[45] Marvin Minsky. *THE EMOTION MACHINE: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster, 2006.

[46] Tom M. Mitchell, Rich Keller, and Smadar Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1(1):47–80, 1986.

[47] K. Murphy, A. Torralba, and W. T. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Adv. in Neural Info. Proc. Systems*, volume 17. MIT Press, 2004.

[48] L. Natale, S. Rao, and G. Sandini. Learning to act on objects. In *Proc. Second International Workshop on Biologically Motivated Computer Vision*. Springer-Verlag, November 2002.

[49] K. Nelissen, G. Luppino, W. Vanduffel, G. Rizzolatti, and G. A. Orban. Observing others: Multiple action representation in the frontal lobe. *Science*, 310:332–336, 2005.

[50] A. Newell, J.C. Shaw, and H.A. Simon. Report on a general problem-solving program. In *Proceedings of the International Conference on Information Processing*, pages 256–264, 1959.

[51] H. Penny Nii. Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, 7(2):38–53, Summer 1986.

[52] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[53] Stephen Peters. *Hyperglue: An Infrastructure for Human-Centered Computing in Distributed, Pervasive, Intelligent Environments*. PhD thesis, Massachusetts Institute of Technology, February 2006.

[54] G.E. Peterson and H.L. Barney. Control methods used in a study of the vowels. *J.Acoust.Soc.Am.*, 24:175–184, 1952.

[55] M. L. Platt and Paul W. Glimcher. Neural correlates of decision variables in parietal cortex. *Nature*, 400:233–238, 1999.

[56] Satyajit Rao. *Visual Routines and Attention*. PhD thesis, MIT, 1998.

[57] R. A. Rensink, J. K. O'Regan, and J. J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8:368–373, 1997.

[58] Charles Rich. Inspection methods in programming. Technical Report AI Lab Technical Report 604, MIT Artificial Intelligence Laboratory, 1981.

[59] Charles Rich and Howard E. Shrobe. Initial report on a lisp programmer's apprentice. Technical Report Technical Report 354, MIT Artificial Intelligence Laboratory, December 1976.

[60] W. Richards. *Natural Computation*. The MIT Press, 1988.

[61] W. Richards and A. Bobick. Playing twenty questions with nature. In Z.W. Pylyshyn, editor, *Computational Processes in Human Vision: An Interdisciplinary Perspective*. Ablex Publishing Corporation, 1988.

[62] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.

[63] Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. L. Erlbaum, Hillsdale, NJ, 1977.

[64] R. E. Schapire and Y. Singer. Improved boosting using confidence-rated predictions. *,Journal of Machine Learning*, 37(3):297–336, 1999.

[65] P. G. Schyns and A. Oliva. From blobs to boundary edges: Evidence for time- and spatial-scale-dependent scene recognition. *Psychological Science*, 5:195–200, 1994.

[66] Howard Shrobe. Dependency directed reasoning for complex program understanding. Technical Report AI Lab Technical Report 503, MIT Artificial Intelligence Laboratory, April 1979.

[67] Howard E. Shrobe, Robert Laddaga, Bob Balzer, Neil M. Goldman, Dave Wile, Marcelo Tallis, Tim Hollebeek, and Alexander Egyed. Awdrat: A cognitive middleware system for information survivability. In *AAAI*. AAAI, 2006.

[68] D. J. Simons and D. T. Levin. Change blindness. *Trends in Cognitive Science*, 1:261–267, 1966.

[69] Push Singh. A preliminary collection of reflective critics for layered agent architectures. In *Proceedings of the Safe Agents Workshop (AAMAS 2003)*, 2003.

[70] Push Singh. *EM-ONE: An Architecture for Reflective Commonsense Thinking*. PhD thesis, Massachusetts Institute of Technology, April 2006.

[71] Push Singh and Marvin Minsky. An architecture for combining ways to think. In *Proceedings of the International Conference on Knowledge Intensive Multi-Agent Systems*, 2003.

[72] Push Singh and Marvin Minsky. An architecture for cognitive diversity. In Darryl Davis, editor, *Visions of Mind*. Idea Group Inc., 2005.

[73] Luca Spalzzi. A survey on case-based planning. *Artif. Intell. Rev.*, 16(1):3–36, 2001.

[74] E. S. Spelke. Initial knowledge: six suggestions. *Cognition*, 50:431–445, 1994.

[75] E. S. Spelke, K. Breinlinger, J. Macomber, and K. Jacobsen. Origins of knowledge. *Psychological Review*, 99:605–632, 1992.

[76] D. Sperber and D. Wilson. Précis of relevance: Communication and cognition. *Behavioral and Brain Science*, 10:697–754, 1987.

[77] Susanne Still and William Bialek. How many clusters? an information-theoretic perspective. *Neural Computation*, 16(12):2483–2506, 2004.

[78] Leo P. Sugrue, Greg S. Corrado, and William T. Newsome. Choosing the greater of two goods: Neural currencies for valuation and decision making. *Nature Reviews*, 6:363–375, May 2005.

[79] Gerald J. Sussman. *A computational model of skill acquisition*. PhD thesis, MIT, Department of Mathematics, 1973.

[80] O. Tchernichovski, P.P. Mitra, T. Lints, and F. Nottebohm. Dynamics of the vocal imitation process: How a zebra finch learns its song. *Science*, 30, March 2001.

[81] D. Thompson. *On Growth and Form*. Dover Publications, 1917, revised 1942.

[82] A. Torralba, K. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *Adv. in Neural Info. Proc. Systems*, volume 17. MIT Press, 2005.

[83] A. Torralba, K. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 2006.

[84] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, 2003.

[85] E.K. Vogel, G.F. Woodman, and S.J. Luck. The time course of consolidation in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance*, in press.

[86] B. C. Williams and P. P. Nayak. A model-based approach to reactive, self-configuring systems. In *Proceedings of AAAI-96*, 1996.

[87] J. M. Wolfe. Visual memory: What do you know about what you saw? *Current Biology*, pages R303–R304, 1998.

[88] Paul J. Zak. Neuroeconomics. *Philosophical Transactions of the Royal Society B (Biology)*, 359(1451):1737–1748, November 2004.