AFRL-HE-WP-TR-2007-0091

Extending the Soar Cognitive Architecture

John E. Laird

University of Michigan
Computer Science and Engineering Division
2260 Hayward Street
Ann Arbor MI 48109-2121

July 2007

Final Report for August 2005 to July 2007

Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
Wright-Patterson AFB OH 45433

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* July 2007	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* August 2005 - July 2007
4. TITLE AND SUBTITLE Extending the Soar Cognitive Architecture		**5a. CONTRACT NUMBER** FA8650-05-C-7253
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) John E. Laird		**5d. PROJECT NUMBER** 63123F
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER DRPA0511
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Michigan Computer Science and Engineering Division 2260 Hayward Street Ann Arbor MI 48109-2121		**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Materiel Command Air Force Research Laboratory Human Effectiveness Directorate Warfighter Interface Division Cognitive Systems Branch Wright-Patterson AFB OH 45433		**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/HECS
		11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-HE-WP-TR-2007-0091

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.
AFRL/PA cleared on 19 September 2007, AFRL-WS-07-2100.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The original purpose of this research was to extend the Soar cognitive architecture based on knowledge gleaned from psychology and brain-based science. Specifically looking at extensions related to memory and learning (episodic, semantic) and emotion. The direction changed when an opportunity became available to collaborate with other biologically-inspired cognitive architecture (BICA) projects to design a completely new BICA called TOSCA. In addition to designing a new architecture, we also designed and implemented a framework that can be used to develop biologically-inspired cognitive architectures. The new framework is called STORM. Finally, throughout the development of TOSCA and STORM we continued the work on developing computational models of emotion. This work continued because even with the design of TOSCA, the exact role of emotion in cognitive architecture remains unclear.

15. SUBJECT TERMS Soar, Biologically-Inspired Cognitive Architectures (BICA), TOSCA, STORM, Cognitive Architecture, Human Cognition, Computational Models

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** John L. Camp
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED	SAR	96	**19b. TELEPHONE NUMBER** *(include area code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

THIS PAGE LEFT INTENTIONALLY BLANK

TABLE OF CONTENTS

List of Figures

List of Tables

Executive Summary

In our original proposal for the DARPA BICA project, our plan was to extend the Soar cognitive architecture based on knowledge gleaned from psychology and brain-based science. We were specifically looking at extensions related to memory and learning (episodic, semantic) and emotion. Our direction changed toward the end of the first year as we saw the opportunity to collaborate with other BICA projects (listed below) to design a completely new biologically-inspired architecture, called TOSCA. In addition to designing a new architecture, we also designed and implemented a framework that can be used to develop biologically-inspired architectures, called STORM. Finally, throughout the development of TOSCA and STORM we continued our work on trying to development computational models of emotion. This work continued because even with the design of TOSCA, the exact role of emotion in cognitive architecture remains murky, in need of further research.

1. Our final report reflects the research we have done under BICA. It has three major parts. The first is an initial design of the TOSCA architecture. Our goal above all else was to be comprehensive – to provide a design for a complete mind that is grounded in the brain. Of necessity in a project this ambitious, our design is still incomplete in many details, but where it is incomplete we have included the known constraints that a final design will have to meet. Although submitted as the final report of the University of Michigan BICA project, this report was prepared in collaboration with the TOSCA team, which is listed below:
   TOSCA Participants (funded under BICA Phase I)
   BICA Thrust A:
   - Michigan (John Laird, Richard Lewis, Thad Polk, Doug Pearson (Three Penny))
   - MIT (Cynthia Breazeal, Linda Smith (Indiana), Larry Barsalou (Emory))

   BICA Thrust B:
   - Dartmouth (Richard Granger, Carey Priebe (Johns Hopkins), Anna Tsao (Algotek))
   - Harvard (Stephen Kosslyn, Giorgio Ganis, Bruce Draper (CSU))
   - Rutgers (Mark Gluck)
2. The STORM framework is a software infrastructure that directly supports expressing the connectivity constraints of the brain, while providing the flexibility to rapidly develop alternative implementations of the functional modules and state variables that make up the a biologically-inspired architecture. The STORM framework was designed and implemented by the Michigan team.
3. Our research on emotion is described in the final part, which is a manuscript under preparation for submission to a journal.

PART I: TOSCA ARCHITECTURE

1. Introduction and overview

Our design of TOSCA starts at the brain system and circuits levels. In developing an initial version of TOSCA, we've chosen to intentionally abstract away from much of the complexity of the brain. Many brain systems include multiple subsystems that are extremely complex in their own right (e.g., vision and hearing within sensory systems) and the sophisticated computational mechanisms underlying these systems are important, but we first need to define the "forest" – the overall architecture with the major pieces and how they fit together, before we get to the "trees". This is purely a tactical decision to get us started and we fully plan to greatly expand the systems and subsystems in TOSCA in the future. Our strategy is to include those neural systems that we consider most important in constructing an initial functional architecture that provides end to end behavior.

The document has the following structure. In Section 2, we identify the innovative claims of our design that distinguish it from other approaches to systems and circuit level models of the brain. Section 3 describes the high-level structure of the architecture in terms of the basic (repeated) architectural loops connecting major cortical and subcortical regions. This basic loop includes perception, categorization (with clustering), access to memories, internal and external actions and action selection, as well as feedback. Section 4 then lays out in more detail the major theoretical commitments concerning the operation of each of the major brain subsystems and their inter-connections, and initial assumptions about representation, time-course, and algorithms. In Section 5, we identify the key emergent functional properties that derive from the integration of the components; many of these properties concern the multi-faceted nature of learning in the system.

2. Innovative claims

TOSCA is *distinguished* by an innovative set of claims concerning the behavior of models built using TOSCA, the computational structures and algorithms that give rise to behavior (the TOSCA architecture itself), and the underlying infrastructure for constructing the architecture (the STORM Framework) that supports direct mappings between TOSCA models and the brain.

Innovative Behavior:

- Learning by Observation
- Other emergent innovative behavior from section 5.

Innovative Architecture:

- Perception:
    - <u>Top-down processing</u> plays a dominant role in perception.
- Categorization:
    - Internal representations are learned by building <u>clusters of sequences of clusters</u> across all sensory modalities. <u>Top-down, predictive processing</u> continues to play a dominant role in categorization.
- Internal knowledge representation, learning, memory, and use:

    - Knowledge is represented as distributed multi-modal structures. These structures can be reinstated or simulated for higher-level prediction and knowledge composition.
- Mental operation selection:
    - Mental (internal) operations use the same basic brain structures as external action selection.
- Action learning:
    - Intrinsic reward based reinforcement learning drives learning of external and internal action selection.
- Integration:
    - By integrating these different learning mechanisms in a complete architecture, the resulting whole should be more powerful than the sum of the parts. Incorporating intrinsic rewards allows useful learning to occur from simple exploration without any explicit task. Learning over mental operations as well as motor actions leads to the development of cognitive skills as well as motor skills. And combining reinforcement learning with clustering over state and action representations makes possible the acquisition of complex skills contingent upon very abstract features.

Innovative Approach to Modeling of the Brain:

- TOSCA is being designed and built using an underlying framework (the STORM Framework) that allows modelers to explicitly declare mappings from state variables and functional modules to regions of the brain. This directly supports analysis of the grounding of the TOSCA architecture and models with the brain and ensures that the connectivity of TOSCA as a computational system matches the known connectivity of the brain.
- The STORM Framework also directly supports declaration and automated management of the time course and activation conditions of functional modules. This simplifies model construction by providing a more abstract substrate for architecture construction and model execution.

3. Overall structure: The basic loop

Our design starts with the human brain, which consists of evolutionarily recent forebrain circuit designs (telencephalic circuits) layered on top of preserved ancient (e.g., reptilian) circuits, with the new designs accounting for more than 90% of the volume of the human brain. There are four primary divisions of telencephalic forebrain (cortex, striatal complex, hippocampal formation, amygdala nuclei), and many subdivisions (e.g., anterior vs. posterior cortex, distinct cortical layers, local circuits, striatal components, hippocampal fields CA1, CA3, dentate gyrus, subiculum, …), each with its own cell types and local circuit design layouts, thus presumably each conferring unique computational properties.

There is (perhaps surprisingly) a single large-scale architecture that organizes all telencephalic components. For almost any given region of posterior cortex, there is a corresponding region of anterior cortex (e.g., the frontal eye fields, connected to posterior visual cortical areas), as well as corresponding regions of striatum, pallidum and thalamus, connected in register. These complementary cortical and subcortical regions are connected in a characteristic pattern: reciprocal connections between posterior and anterior cortex, converging anterior and posterior cortical projections to a related region of striatum, which in turn connects (via pallidum and thalamus) back to the same region of anterior cortex. This overall "systems circuit" is by far the largest coherent loop in the mammalian brain, and it is repeated for multiple regions of posterior cortex, with dedicated regions corresponding to individual sensory modalities, as well as non-cortical telencephalic regions including components of hippocampus and amygdala, connected with dedicated regions of striatum and anterior cortex.

We are faced with a difficult problem in describing the design of TOSCA. The operation of a specific component is important, but the interaction among components is equally (or even more) important. Moreover, we have interactions between groups and loops of components. Our approach is to initially focus on the basic loop of behavior from perception to action and describe the primary neural systems that participate in that loop. This will leave out some structures that play a less central role in the basic loop. In going through the loop we often give a cursory description of a component because our goal is to build up the big picture, emphasizing interactions.

At the highest level, the initial version of the TOSCA architecture will attempt to tightly integrate the most important neural systems found in the brain. These neural systems are described in detail in Section 4, and are labeled below with their corresponding subsection. The descriptions in parentheses summarize the main computational functions we attribute to each system.

4.1. Sensory systems (Low-level vision and audition)
4.2. Specific thalamocortical (core) circuits (Clustering)
4.3. Non-specific thalamocortical (matrix) circuits (Sequencing)
4.4. Cortico-cortical circuits (High-order vision, state-intention associations, intention-state associations)
4.5. Cortico-hippocampal circuits (Episodic memory)
4.6. Cortico-striatal circuits (Intention selection)
4.7. Dopamine reward circuits (Reinforcement learning)
4.8. Cortico-amygdala circuit (Emotion, State-dependent storage & retrieval)

Figure 1 presents some of the major pathways among the structures involved in these neural systems. Many others that play a less central role in the basic operation of TOSCA are omitted. In contrast to many traditional cognitive architecture where functionality maps directly to architectural components, in TOSCA (and the brain), it is the circuits and loops through multiple neural structures from which functionality emerges.

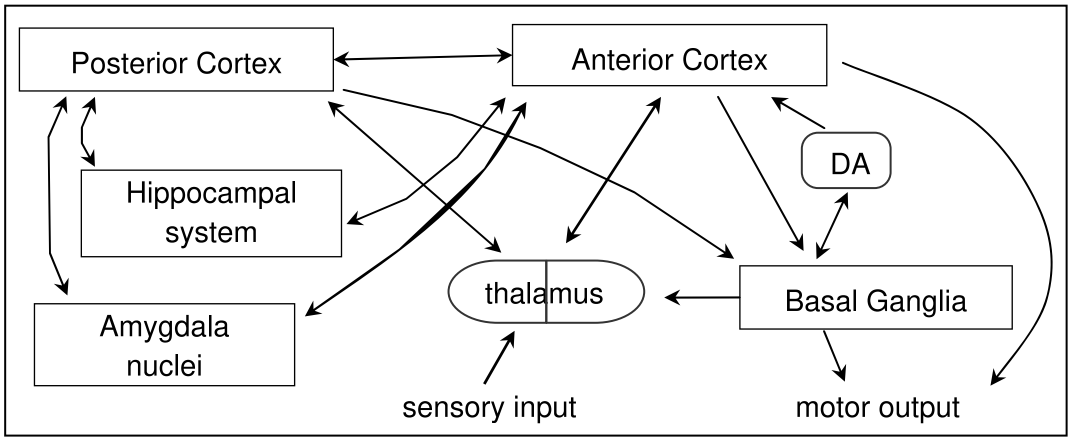


Figure 1. Major pathways within and between neural systems underlying TOSCA

Based on the specific functions we attribute to each of the major neural systems and on the pattern of connections in Figure 1, we propose that the operation of the human neural architecture can best be understood in the familiar concepts of reinforcement learning: states, actions, values, and rewards. At any given point in time, the architecture has a representation of the current **state** (in cortex). The state includes information about the external environment (delivered by perception in different modalities), information retrieved from memory (e.g., from the hippocampal system), information about emotional state (from the limbic system including the amygdala), information about current goals, and additional information derived from cortico-cortical associations that augment or elaborate the state (e.g., inferences about other aspects of the state that are not directly perceived or remembered).

Based on the current state, the architecture must decide what to do next. In standard reinforcement learning models, this decision corresponds to choosing the next action to take. However, the term action is strongly associated with motor behavior, and we assume that the neural architecture often chooses mental actions, rather than motor actions. We therefore adopt the more neutral term **intention** rather than action, and talk about selecting an intention rather than selecting an action. We further assume that intentions can correspond to high-level goals that are not immediately achievable but that influence subsequent behavior (e.g., deciding to go to Hawaii). Again, the term intention seems more consistent with this connotation than does the term action.

We assume that intentions are represented in anterior cortex and that cortico-cortical projections to anterior cortex propose specific intentions based on the current state. The strength of the association between a given state and intention corresponds to the current **value** associated with choosing that intention in the context of that state. When multiple, conflicting intentions are proposed, the architecture must make a selection and we assume that corticostriatal loops provide this intention-selection capability.

When a *motor* intention is selected, the motor program associated with that intention is executed. The execution of a *mental* intention, on the other hand, corresponds to making a change to the current state representation in cortex. Mental intentions could include a wide range of cognitive mechanisms, such as selective attention, working memory, inhibition, and imagery.

To complete the reinforcement learning loop, we need to update the value of state-intention associations when a selected intention leads to more (or less) **reward** than expected. We assume that the mesocortical dopamine system serves this function. Specifically, the activity of midbrain dopamine neurons reflects the error in predicted future reward. If executing an intention leads to a state that is better than expected, then midbrain dopamine neurons fire at an increased rate, releasing dopamine in anterior cortex, and potentiating synapses onto the active intention representation. This potentiation makes it more likely for that intention to be selected in similar future states and corresponds to increasing the value of the state-intention association. We assume that a small number of *intrinsic* rewards are built in to the architecture and allow it to learn from simple exploration even when there are no explicit external rewards, providing a means to bootstrap to more sophisticated learning.

In addition to learning the values associated with state-intention associations, the proposed architecture will also constantly be learning about regularities in the state. Specifically, we assume that thalamocortical circuits implement algorithms that cluster over input regularities and that learn to recognize and predict frequently occurring sequences of such clusters. Furthermore, the hierarchical structure of cortex will lead to recursion: downstream cortical areas will cluster over sequences of clusters in upstream areas which in turn will cluster of sequences of clusters in cortical areas further upstream. The result is a very rich hierarchical state representation consisting of clusters of sequences of clusters of sequences.

Another critical type of learning in the architecture is carried out by the episodic memory system in the hippocampal system. We assume this system is constantly encoding critical aspects of the state representation and supporting retrieval of similar episodes from past memory. This kind of memory makes it possible for the system to make better predictions about the results of actions it has taken (or seen other agents take) in the past and to do so quickly. Such fast learning provides an important complement to TOSCA's other learning mechanisms (e.g., clustering, reinforcement learning) which are good at exploiting regularities but take longer to do so.

Our hypothesis is that the combination of these mechanisms will produce an architecture with substantially more power and flexibility than existing alternatives. Incorporating intrinsic rewards allows useful learning to occur from simple exploration without any explicit task. Learning over mental operations as well as motor actions leads to the development of cognitive skills as well as motor skills. Combining reinforcement learning with clustering over state and action representations makes possible the acquisition of complex skills contingent upon very abstract features. And incorporating episodic learning makes it possible to benefit quickly from past mistakes and successes before slower learning mechanisms can.

4. Major brain subsystems

This section describes the major brain subsystems that will be modeled by TOSCA. These are not isolated regions of the brain but are instead circuits involving multiple brain regions. For each of these subsystems we first present the underlying anatomical structures in the brain, with an accompanying figure that highlights those structures in Figure 1 that contribute to the circuit. This is followed by a description of the physiological operation of the structure. We then discuss the derived computational functionality – this is the core of what we will implement in software. This is followed by a description of how this subsystem interacts with the rest of the system.

4.1 Sensory Processing

Anatomical structure

Sensory systems in the human brain combine specialized cortical areas with more general cortical-subcortical loops. The early visual system is among the most intensively studied subsystems of the brain, and includes the retina, dorsal lateral geniculate nucleus (LGNd) of the thalamus, superior colliculus of the midbrain, and cortical areas V1 through V4. The early auditory system consists of the cochlea, auditory brainstem nuclei, the medial geniculate nucleus of the thalamus, and the primary auditory cortex (A1).

The early visual areas are unique in that they are "retinotopically mapped". This means that (1) the majority of afferent inputs to a cell are projections from a well-defined receptive field on the retina, and (2) neighboring cells usually respond to neighboring receptive fields. As a result, the 2D topographic structure of the retinal image is preserved throughout the early visual system. This 2D structure is also exploited by top-down processing for mental imagery and by tactile sensing when reading Braille. Not all of vision is retinotopic, however. Most visual processing occurs in later specialized cortical areas within the vision system where data is not so image-like.

While the TOSCA project emphasizes the critical role of general-purpose cortical/subcortical loops, specific cortical regions and pathways also assume specialized functional roles within the architecture of the brain. Cortical regions are particularly important in vision, which dominates the posterior cortex and involves the anterior cortex as well.

Areas V1 through V4 are at the heart of the early visual system. After these areas, the vision system divides into the ventral and dorsal pathways. Roughly speaking, the ventral (or "what") pathway processes object properties, while the dorsal (or "where") pathway processes spatial properties. The ventral pathway includes portions of the lateral occipital cortex (including but not limited to areas V7 and V8) and the anterior infero-temporal cortex. The dorsal pathway includes the medial-temporal (MT) cortex (sometimes called V5) and large portions of the posterior parietal cortex. Both paths connect spatially adjoining cortical regions, allowing for strong cortical-to-cortical connections. Information from the ventral and dorsal pathways converge in the anterior cortex in the angular gyrus and Area 19.

Physiological operation

The physiology of the early visual system has been studied more extensively than any other part of the brain. We know, for example, that information is divided into the α and β channels as early as the retina. By the time it reaches the visual cortex, there are separately identifiable maps for edge orientation, color, disparity and motion. The responses of cells within the edge orientation maps have been particularly well studied. The bottom-up responses of so-called "simple cells", for example, can be mathematically modeled as Gabor filters of the cell's receptive field; the responses of the far more numerous "complex cells" can be modeled as Gabor energy responses combined across phases. There is also a litany of other, less common cell types within the edge maps, for example end-stopped cells and grating cells.

So much attention has been paid to the bottom-up responses of cells in the early visual system that its larger role within the human brain is sometimes ignored. In particular, the role of top-down processing in the early vision system is insufficiently emphasized. There is an order of magnitude more feed-back projections from visual cortex back to thalamus than feed-forward projections from thalamus to visual cortex, strongly suggesting the powerful role played by top-down cortical modulation. Predictive spatial attention can be directly measured at the neural level in the visual cortex (and perhaps even LGNd) by increased baseline firing rates in selected locations, occurring before the predicted stimulus is presented. Perhaps most importantly, the traditional models of cells in early visual cortex (e.g. Gabor filters) describe their response within the first 80ms to stimuli presented out of context. After 80ms, efferent projections become active and the responses of cells in V1 become considerably more complex.

Audition is a lot like vision. There are more top-down than bottom-up connections between A1 and the thalamus, and top-down processing and gating are fundamental to the sense of hearing. Rather than a view in which signals pass from the periphery into cortex, the system is instead a highly active perceiver, controlling and predicting inputs throughout the perceptual process and directing the paths by which inputs arrive.

The interactions among the six major cortical components of human vision are shown in Figure 2. The visual buffer represents the early, retinotopic stages of visual processing. The object properties subsystem represents the ventral visual stream. Its role is to group familiar stimuli into view-specific categories; in essence, to match the current stimulus to stimuli it has seen before. The spatial properties system represents the dorsal stream, and is concerned with the relative 3D positions and motions of objects, tracking, and hand-eye coordination.

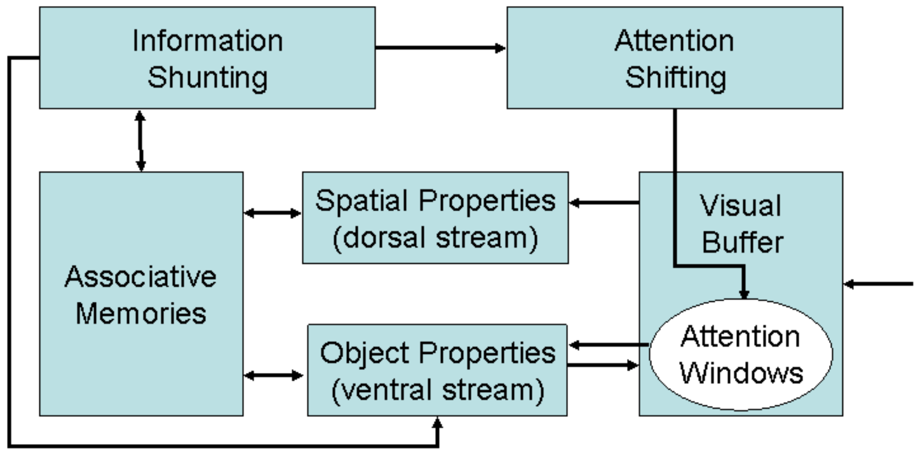


Figure 2. Function components of human vision

Just as important are the anterior cortical structures on the left of Figure 2. Associative memory is often omitted from diagrams of human vision, but it is the target of both the ventral and dorsal visual streams. The associative memories and information shunting systems are responsible for bridging the gap from recognition, defined as knowing you have seen an object before, to identification, which is defined as the ability to retrieve multi-modal information about an object, such as its name, what it feels like, and the sounds it makes.

Associative memories and information shunting also provide top-down visual predictions. People are able to recognize objects completely out of context, but that is a rare scenario found mostly in psychology labs. Most of the time, we recognize objects in place. We know what to look for because we have been in a location before and remember what we saw, or we rely on semantic memory to predict what we are likely to see in a new location, based on "gist", scene type and context. Visual perception is largely a process of confirming predictions, with bottom-up vision recognizing unexpected objects and boot-strapping learning and memory.

Derived computational functionality

From a whole-brain perspective, spatial attention is not just one role of the early visual system: it is its primary function. Most visual processing after the early vision system is restricted to a limited set of attention windows. As a result, it is more efficient to compute stimulus features after the early vision system, when they only have to be computed over the selected windows. The feature maps in the early vision system, therefore, are largely computed where they are because they are needed for selective attention. (The exceptions are motion features, which in addition to contributing to selective attention are also used extra-attentionally by the parietal visual stream for ego-motion estimation and related tasks)

Selective attention serves at least three purposes for the vision system. It is a data reduction technique that limits the amount of visual information that has to be processed downstream. More importantly, selective attention provides translational and scale invariance for object recognition, by creating local coordinate systems centered on fixed locations. Most importantly of all, selective attention is a key mechanism for integrating top-down and bottom-up processing; it allows the visual system to balance its efforts between confirming predictions and interpreting unexpected or novel stimuli.

The functionality of the visual cortical areas can be divided along the lines of the two major pathways. The ventral stream is concerned with object recognition. It matches current stimuli to previous stimuli, at a rough categorical level. In the case of familiar objects, it also extracts subcategorical information, for example the gender, age and expression of human faces. It does so in a manner that is independent of translation and scale, but not necessarily independent of planar rotation, perspective or illumination. The dorsal stream is concerned with depth, motion and tracking. It recovers the approximate position of objects that are within arms' reach, and qualitative depth information for more distant objects. It detects and estimates ego-motion, and tracks independently moving objects.

Both visual streams work by integrating top-down and bottom-up processing. While bottom up visual processes have been studied extensively, top-down vision is less well understood. We believe it can be divided into *strategic* and *reflexive* top-down processing. Strategic processing is driven by the higher-level goals of the agent, as in visual search tasks. It can also be used to explicitly differentiate among two or more competing visual hypotheses. Reflexive top-down processing, on the other hand, is driven by automatic predictions generated on the basis of learned statistical regularities. Early on, for example, a toddler might learn that human faces include two eyes, a nose and a mouth; from then on, recognizing any one of these parts might generate reflexive predictions for the other components.

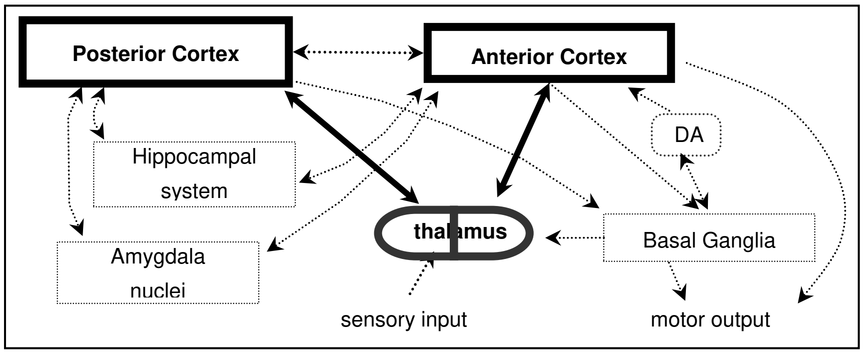
Another distinction in top-down vision is the difference between *threshold lowering* and *filling in*, two terms that can be thought of in terms of the classic signal detection concepts of **d'** and **β**. In threshold lowering, an agent is willing to make a decision based on degraded information. To borrow an example from David Marr, a gardener sees a patch of brown dart toward a cabbage patch, and immediately believes it's a rabbit. In this scenario, there is actually very little signal to base this decision on, but given the strong contextual constraints and the inability to get more data, the gardener makes a snap judgment by lowering the decision threshold top-down. In terms of signal detection theory, top-down vision has lowered the threshold **d'** needed to reach a decision. Alternatively, top-down processing may be used to fill in more information to distinguish among competing but similar hypotheses, effectively increasing **β.** To push Marr's example further than he intended, rabbits can be discriminated from hares based on the size of the ears and hind legs. If the animal stays in view, an animal expert might distinguish the two hypotheses by filling in these hypotheses.

Systems

Most of the "heavy lifting" in visual understanding is carried out in the ventral and dorsal streams, not the early vision system, and via "top-down" modulatory control by cortical structures of early sensory areas. The TOSCA team brings extensive longstanding background expertise on the primacy of top-down processing in vision. The unique role of the early visual system is to compute features across the retina that are needed for selective attention, to modulate the cross-cutting currents of top-down and bottom-up vision.

4.2 Specific thalamocortical (core) circuits (clustering)

Anatomical structure



Projections from cells in thalamic "core" nuclei synapse on target neurons in all cortical layers to some extent but predominantly in deep layer III and (in granular cortical regions) in layer IV, as well as on the apical dendrites of layer VI neurons. These afferents, which preserve topographic organization, are often described as the primary input to sensory neocortical regions, though quantitative neuroanatomical studies report that these thalamic inputs constitute a very small percentage of the total set of afferents to cortical layer IV cells: for instance, inputs from dorsal lateral geniculate nucleus (dLGN) neurons comprise less than 6% of the synaptic contacts onto layer IV target cells in primary visual cortex. Projections from a given thalamic core region extend to a cortical area roughly 0.5 – 1.0 mm wide, somewhat larger than the size of physiologically delineated functional columns. Layer VI axons project back topographically to the thalamic core cells from which they receive inputs, as well as to the overlying portion of the nucleus reticularis (NRt) covering the target core cells. NRt in turn generates GABAergic projections to these thalamic core cells. The result can be depicted by highlighting a subset of the connections that occur in thalamocortical circuits as in Figure 3.

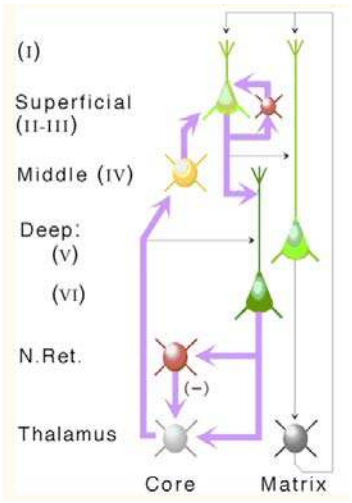


Figure 3. Thalamocortial Circuits

Physiological operation

Here we describe the simplified steps that occur in response to normal inputs. Peripheral inputs activate thalamic core cells which in turn participate in topographic activation of middle cortical layers, e.g., ear → cochlea → auditory brainstem nuclei → ventral subdivision of medial geniculate (MGv), or corresponding thalamic core nucleus ("core" in the figure), → primary auditory cortex (A1), layer IV → layer II-III → layer VI → N.Ret → MGv (core).

In the event that a fixed input is being focused on, i.e., that a stream of inputs is not arriving, then this loop will recur. (In the next section, it will be seen that in response to a stream of inputs, other thalamocortical mechanisms become engaged, interrupting operation of the core loop and employing elements of both core and matrix loops).

As the core loop recurs in response to fixed input stimuli, a series of physiological responses occurs. The superficial (layer II-III) cells that are most response to a given input will in turn activate neighboring inhibitory (red) cells, which then in turn inhibit all the excitatory cells in the region. Thus the response to an input is a relatively brief discharge from only the most responsive excitatory cells followed by silence induced by lateral inhibition. As the synaptic contacts onto the responding cells become strengthened via LTP, those cells become increasingly probable responders even to slightly different spatial input patterns. Thus those superficial cells that initially respond to a particular input pattern become increasingly responsive not only to that input but also to a range of similar inputs (those that share many active lines; i.e., small Hamming distances from each other), such that similar but distinguishable inputs will come to elicit identical patterns of layer II-III cell output, even though those inputs would have given rise to slightly different output patterns in the absence of LTP.

These learning-based (LTP-based) effects can be simply characterized in terms of the formal statistical operation of clustering, in which sufficiently similar inputs are placed into a single category or cluster. This is further discussed in the next section, on functional implications.

Immediately following this response from superficial layer neurons, those cells activate deep layers (V and VI; see Figure 3). Output from layer VI initiates feedback activation of nucleus reticularis (N.Ret) (Liu and Jones 1999) which in turn inhibits the core thalamic nucleus. Since, as described above, topography is preserved through this sequence of projections, the portions of the core nucleus that become inhibited will correspond topographically to those portions of L.II-III that were active. On the next cycle of thalamocortical activity, the input (assumed as above to be a relatively fixed unchanging input) will arrive at the core nucleus against a background of inhibitory feedback from N.Ret, which has been shown to last for hundreds of milliseconds (Cox et al., 1997; Zhang et al., 1997). Thus the predominant component of the next input to cortex is only the uninhibited remainder of the input, whereupon the same operations as before are performed. Thus the second cortical response will consist of a quite distinct set of neurons from the initial response, since many of the input components giving rise to that initial response are now inhibited. This process of inhibition and distinct selected responses continues until the feedback inhibition at N.Ret diminishes (roughly 500 – 1000 msec).

Derived computational functionality

Analysis of the sequence of responses in computational models has shown clustering and successive sub-clustering of inputs. The first cycle of response identifies the input's membership in a general category of similar objects (e.g., flowers); the next response (a fraction of a second later) identifies its membership in a particular subcluster (e.g., thin or fat flowers); then sub-sub-clusters, etc. Thus the system repetitively samples across time, differentially activating specific target neurons at successive time points, to discriminate among inputs (see, e.g., Kilborn 1996; Rodriguez et al., 2004).

Table 1: Simplified Thalamocortical Core Algorithm

```
for input X
for C ∈ win(X,W)
W_j ⇐ W_j + k(X − C)
end_for
X ⇐ X − mean(win(X,W))
end_for
where
X = input activity pattern (vector); W = layer I synaptic weight matrix;
C = responding superficial layer cells (col vector); k = learning rate parameter;
win(X,W) = column vector in W most responsive to X before lateral inhibition [∀j, max(X · W_j) ]
```

The method can be characterized as an algorithm (Table 1). Analysis reveals the algorithm's time and space costs. The three time costs for processing of a given input X are: i) summation of inputs on dendrites; ii) computation of "winning" (responding) cells C; iii) synaptic weight modification. For n learned inputs of dimensionality N, in a serial processor, summation is performed in O(nN) time, computation of winners takes O(n) time, and weight modification is O(N log n). With appropriate parallel hardware, these three times reduce to O(log N), O(log n), and constant time respectively, i.e., better than linear time. Space costs are similarly calculated: given a weight matrix W, to achieve complete separability of n cues, the bottom of the constructed hierarchy will contain at least n units, as the leaves of a tree with log Bn hierarchical layers, where B is the average branching factor at each level. Thus the complete hierarchy will contain ~ n[B/(B-1)] units, i.e., requiring linear space to learn n cues (Rodriguez et al., 2004).

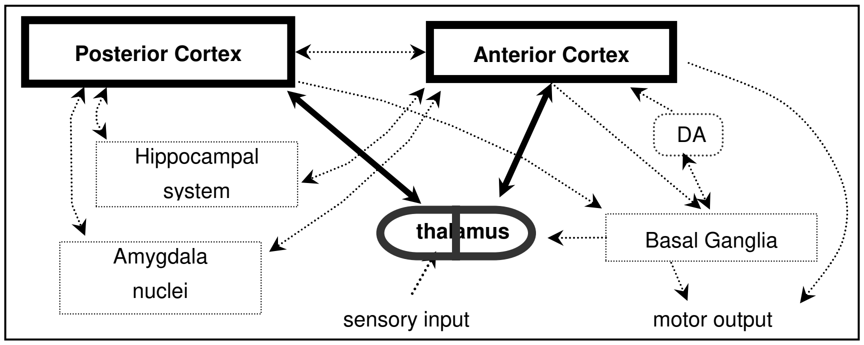
These costs compare favorably with those in the (extensive) literature on such methods (Rodriguez et al., 2004). Elaboration of the algorithm has given rise to families of computational signal processing methods whose performance on complex signal classification tasks has consistently equaled or outperformed those of comparable methods (Coultrip and Granger, 1994; Kowtha et al., 1994; Granger et al., 1997; Benvenuto et al., 2002; Rodriguez et al., 2004).

Systems

The thalamocortical core loop is part of the overall thalamocortical loop, which includes the matrix circuit, discussed in the following section. Taken together, thalamocortical loops are the primary circuit in the brain, engaged in every cortical region, and in turn participating in cortico-striatal and cortico-limbic (hippocampal and amygdala) circuits, each described later.

4.3 Non-specific thalamocortical (matrix) circuits (sequencing)

Anatomical structure



Projections from cells in thalamic "matrix" nuclei predominantly connect in layer I, chiefly on the apical dendrites of neurons from layers II, III, and V. These projections have been referred to as "nonspecific," in i.e., broad and diffuse in contrast to the more topographic projections from "core" nuclei in thalamus (Lorente de No 1938; Killackey and Ebner 1972, 1973; Herkenham 1986; Jones 1998). It has consistently been confirmed that matrix cells projecting to a given cortical area receive projections back from layer V of that cortical area without intervening NRt contacts (Conley and Diamond 1990; Rouiller et al 1991; Bourassa and Deschenes 1995; 1998). This portion of the thalamocortical circuit can be illustrated by highlighting a subset of the connections that occur in thalamocortical circuits as in Figure 4.

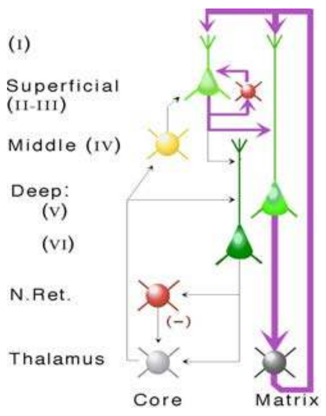


Figure 4. Thalamocortical Matrix Circuit

Physiological operation

Unlike the core loop, the matrix circuit receives no inputs from peripheral signals. Only after cortex is activated by inputs via the core loop, is the matrix loop activated. In particular, once the superficial layer cells in cortex respond to an input, their output activates not just layer VI as described in 4.2 but also layer V, which sends diffuse (non-topographic) feedback to matrix nucleus Mt, which in turn projects back up to layer I of cortex. Their non-topographic nature means that these projections do not retain any neighbor relations that may obtain among inputs.

(This loop, like that of the core circuit, is timed via endogenous "clocks": synchronous activity of wide regions of cortex (modulated in part by ascending systems affecting the periodic responsivity of inhibitory cells) makes the probability of excitatory cell spiking lower during peak inhibition and higher during inhibitory troughs. Moreover, the average time course of excitatory and inhibitory potentials (15 msec and 50 msec, respectively), and the time constants of dendrites, severely limit the temporal precision with which a target neuron can "read" differences among slightly different spike trains (Magee 2000).)

Cortical pyramidal cells preferentially respond to onsets and offsets, i.e., transitions among inputs in all cortical areas studied (somatosensory: Peterson et al. 1998; auditory: Recanzone et al. 2000; visual: Rols et al 2001; Bair et al 2002).

The activation of layer V in rapid sequence via activation by superficial layers (in response to each element of a sequence) and via activation by Mt (corresponding to feedback from previous element in the sequence) selects responding cells sparsely from the most activated cells in the layer (Coultrip et al., 1992) and selects synapses on those cells sparsely as a function of the sequential pattern of arriving inputs. Thus synapses potentiated at a given step in layer V correspond to the input occurring at that time step together with orthogonalized feedback arising from input just prior to that time step (Aleksandrovsky et al. 1996; Rodriguez et al. 2004).

Derived computational functionality

The same steps as those described in section 4.2 obtain, but in response to time-varing inputs, a different effect is produced: that of "chaining" the elements in the input sequence via the "links" created due to layer V activity from coincident inputs corresponding to current and prior input elements. As in the operating rule described by Granger et al. (1994), the sparse synaptic potentiation enables the cells in layer V to act as a novelty detector, selectively responding to those strings that have previously been presented. Whereas superficial layer cells in the model respond to any of a number of sufficiently similar inputs (the "clustering" effect described earlier), the deep layer cells respond only to the input sequences that have actually occurred previously, due to the orthogonalizing input from Mt combining with superficial layer input. Thus the layer V activation patterns even for very similar input sequences will be very different from each other, or, put differently the probability of two similar input sequences eliciting similar sequences of layer V patterns is low.

Table 2: Simplified Thalamocortical Matrix Algorithm

```
for input sequence X(L)

for C $\in$ TopographicSuperficialResponse(X(L))

for V(s) $\in$ C $\cap$ NNtResponse(X(L-1))



Potentiate( V(s) )

NNt(L) $\Leftarrow$ NontopographicDeepResponse(V)
end_for
end_for
end_for
```

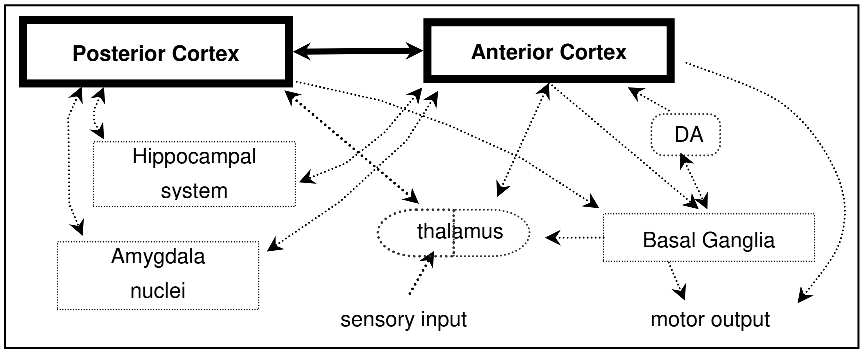
where L = length of input sequence;
C = columnar modules activated at step X(L);
V(s) = synaptic vector of responding layer V cell,
NNt(L)= response of nonspecific thalamic nucleus to feedback from layer V.

As before, the method can be characterized as an algorithm (Table 2). Rodriguez et al. (2004) showed that the space costs grow linearly with the number of sequences stored, for an assumed fixed acceptable rate of collision errors.

Systems

The thalamocortical loops are part of the overall cortico-cortical and cortical-subcortical systems-level organization of the telencephalic model. The primary representations, hierarchically nested sequences of categories, are elaborated in various ways via these interactions. As will be seen in Section 5, these representations underlie content all the way from perception to language.

4.4 Cortico-cortical circuits (state-intention associations, intention-state associations)



Anatomical structure

We have already discussed a number of aspects of the anatomy of cortex that will be incorporated in TOSCA: specific assumptions about the anatomy of the sensory system including cortical specialization (section 4.1) and about the microcircuitry within cortical columns and interactions with thalamic nuclei (sections 4.2-4.3). In this section we focus on two other higher-level characteristics of cortical anatomy that play an important role in the operation of TOSCA: (1) massive, bidirectional connectivity within and between cortical areas particularly into and out of anterior cortex, and (2) topographic projections: nearby cells tend to project to nearby targets.

Physiological function

Distinct cortical areas perform quite different functions. At the highest-level, sensory information is processed in posterior neocortex (consisting of occipital, parietal, and temporal neocortex) while anterior cortex (frontal cortex) is primarily involved in processing actions and intentions. At a finer grain-size, there are on the order of 50-100 distinct cortical areas (or more depending on how you divide it up) performing quite different functions. Many of these cortical areas are organized topographically with nearby cells exhibiting similar receptive fields. Cortical representations of multimodal stimuli/concepts involve a large population of active neurons distributed across multiple cortical areas. Processing across cortical areas is strongly interactive as the activity within a cortical area can be strongly influenced both by bottom-up influences (e.g., perceptual processing in sensory cortex) and top-down influences (e.g., attentional influences from anterior areas).

Computational function

At the most coarse level, TOSCA assumes that cortex contains a representation of the current state and of potential intentions. Intentions are specifically assumed to be represented primarily in the frontal lobes (anterior cortex). Cortico-cortical projections into anterior cortex will encode associations between specific features of the state and specific intentions. Conversely, efferent projections from anterior cortex to other cortical areas will encode a mental operation by specifying how a given intention should change the state. Simple examples include exciting part of the posterior state representation in order to maintain it (working memory), focusing attention on some particular feature of the state, activating or manipulating a mental image in sensory cortex, among many others. Of course, projections within posterior cortex are also capable of changing state. We assume these projections correspond to better learned, more automatic associations whereas frontal representations correspond to more controlled, deliberate intentions.

As previously discussed, the central representation in TOSCA will be sequences of clusters (of sequences of clusters...). Clusters naturally arise from thalamocortical loops which perform a kind of competitive learning: neurons whose receptive field best matches the current input "win" and their receptive field is modified to be closer to that input. This is a standard approach in connectionist modeling and is known to lead to receptive fields that represent category prototypes.

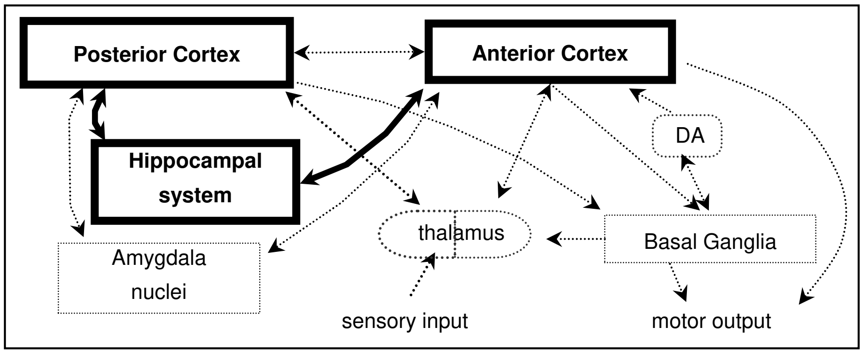
We adopt the additional assumption that learning affects the receptive fields of neurons that are spatially near the "winning" cells. This assumption is quite plausible under the assumption that nearby cells tend to cooperate (e.g., excite each other) rather than compete. This is the critical assumption underlying all self-organizing map (SOM) models and leads to the kind of topographically organized networks that are ubiquitous in cortex. What this means for TOSCA is that clusters that are represented nearby in cortex will tend to represent similar stimuli/concepts and will tend to project to nearby targets. Such topographic organization naturally supports similarity-based generalization under the assumption that cortical representations correspond to population codes rather than grandmother cells. To see this, consider what happens when an association is learned between one cluster corresponding to a feature of the state and another cluster corresponding to a potential action to take when that feature is present. Each representation corresponds to a large population of nearby cells with similar receptive fields (a population code). Learning the association between them corresponds to strengthening the connections between the two populations. Similar, but slightly different, features of the state will activate an overlapping population of cells as the original feature and, as a result, the new features will be partially associated with the same action. As a result, the architecture will be able to choose actions that are generally appropriate based on states it has never experienced, as long as those states are similar to states it has experienced.

Systems

The cortical system will interact with all the other major subsystems in TOSCA: thalamus (clustering and sequencing), hippocampal system (episodic memory), corticostriatal circuits (action selection), midbrain dopamine system (reinforcement learning), amygdala (reciprocal priming). These interactions are described in the other parts of section 4.

4.5 Cortico-hippocampal circuits (episodic memory, spatiotemporal relations)

Anatomical structure



As illustrated below in detail (Figure 5), our network model of cortico-hippocampal circuits for learning and memory will include modules corresponding to the dentate gyrus (DG), CA3 and CA1 fields of the hippocampus proper, and superficial and deep entorhinal cortex, which receive inputs from the perihinal and parahippocampal coritices which, in turn, get projections from rest of the brain.

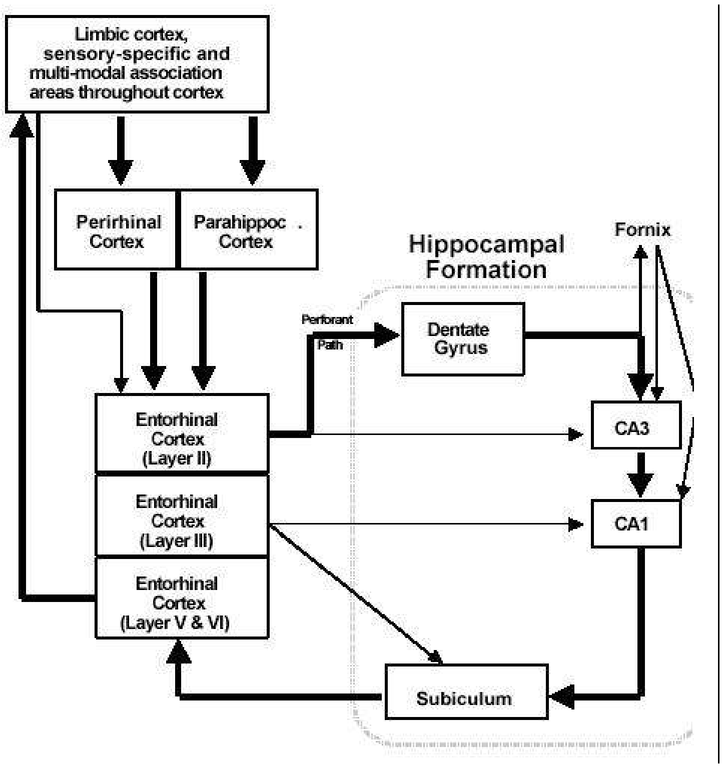

Limbic cortex, sensory-specific and multi-modal association areas throughout cortex
Perirhinal Cortex
Parahippoc . Cortex
Hippocampal Formation
Fornix
Perforant Path
Dentate Gyrus
Entorhinal Cortex (Layer II)
Entorhinal Cortex (Layer III)
Entorhinal Cortex (Layer V & VI)
CA3
CA1
Subiculum

Figure 5. Hippocampal Formation interaction with Cortex

Entohrinal Cortex. The entorhinal cortex contains six layers that, for simplicity, can be divided into "superficial" (layers I-III) and "deep" (layers V-VI) EC. The superficial layers receive highly-processed multimodal sensory input from neocortex (primarily via perirhinal and postrhinal cortex). Principal neurons in the superficial layers include pyramidal neurons (in layer III) and stellate cells (in layer II). The stellate cells project via the perforant path to DG and CA3, while the pyramidal cells project to CA1 (and subiculum). The superficial layers also contain a large number of GABAergic interneurons that exert a widespread inhibitory control over the output of principal cells. The deep layers receive input from CA1 (and subiculum) and project back to the same neocortical areas that provided input to the superficial layers. There is also a projection from deep to superficial EC that causes both excitation and feedforward inhibition (van Haeften et al., 2003). Pyramidal cells in the deep layers show graded persistent firing (over 5 minutes) which could allow for reverberating circuits (superficial EC to hippocampus to deep EC to superficial EC) to maintain stimulus representations across short delays (Frank & Brown, 2003).

Hippocampal Formation. The hippocampus includes a dentate gyrus (DG) layer, a CA3 layer, and a CA1 layer. Connections from DG to CA3 and from EC to CA1 are topologically organized. Each stellate neuron in EC contacts a subset of the possible postsynaptic targets in DG and in CA3. Each neuron in CA3 contacts a subset of the possible postsynaptic targets in CA3 and CA1.

Physiological function

EC neurons receive external input representing highly pre-processed multimodal sensory information from cortex. They will be modulated by interneurons providing both feedback and feedforward inhibition; for simplicity. Strong inhibitory processes and local circuit feedback in the EC cause representational compression, implementing representational clustering function proposed by Myers et al. (1995). Deep EC neurons form the principal output of the hippocampal region back to cortex and also project to principal cells in superficial EC.

Computational function

In our implementation of TOSCA, we will follow the widely accepted hypothesis that the hippocampal region plays a critical role in the acquisition of new memories, both (1) rapidly-acquired memories for autobiographical events, sometimes collectively called **episodic memory** (e.g. Squire, 1987; Squire et al., 2004), as well as being critically involved in developing novel **adaptive stimulus representations** that are important both for episodic memories but also for **incrementally-acquired procedural memories** which are otherwise mediated through the cerebellum and basal ganglia. As a starting point we plan to incorporate our previous neural network modeling of hippocampal region processing in the larger architecture (Gluck & Myers, 1993, 2001; Myers & Gluck, 1994). This model assumes that the hippocampal region develops new stimulus representations that encode contextual and stimulus-stimulus regularities. Specifically, we found that known features of the anatomy and physiology of EC (sparse activation of principal neurons, dense inhibition, and local plasticity mechanisms) give rise to the compression of redundant features in the input. This model accounted for data showing that latent inhibition and sensory preconditioning, which depend on compressing together the representations of CS and context and/or co-occurring cues, survive selective hippocampal lesion but are impaired after EC or broad hippocampal-region damage (Myers et al., 1995). We will adopt this same model in the initial version of TOSCA. We will also follow our previous modeling in assuming that the hippocampal layer forms a compact code for the whole situation in which the organism finds itself (what we call the 'ensemble"; Murnane, Phelps, & Malmberg, 1999). Such representations will form the basis of episodic memory in TOSCA, which are acquired in one or a few exposures and include information about the spatial and temporal context in which learning occurred (e.g. Meeter et al., 2004; Hasselmo & Eichenbaum, 2005; O'Reilly & Rudy, 2000), or on spatial and sequence learning, which may be animal analogues of human episodic learning (e.g. Lisman et al., 2005; Sharp, 1999; Tsodyks et al., 1996).

Systems

Interactions between the hippocampal system and other neural systems will play a crucial functional role in TOSCA. At the highest level, the hippocampal system will constantly be encoding and storing compressed representations of the current state (as represented in posterior cortex). When similar states are encountered in the future, they will activate the previously stored compressed representation, which will in turn reinstantiate information from the previously stored state in posterior cortex. Once this information is represented in posterior cortex, it can influence which actions/intentions are proposed and selected. Furthermore, we envision corticohippocampal loops in TOSCA storing and retrieving temporal sequences of events that have been experienced. Specifically, each event in a sequence could provide cues that lead to retrieval of the next event in the sequence. In this way, the hippocampal system could be used to replay a sequence of events from the past. Doing so could be potentially very valuable to the agent, because it would make it possible to plan ahead and predict likely future events that may improve its present decision making.

The interaction between the hippocampal system and anterior cortex could provide another crucial functionality for TOSCA. Recall that one critical assumption of the architecture is that it learns how and when to perform mental operations as well as motor actions. That is, the same learning algorithms will be used to reinforce rewarding actions, whether they are mental actions or motor actions. The initial design of TOSCA will exploit this strategy in order to learn how best to exploit its episodic memory system. For example, TOSCA should be able to learn when the mental act of attempting an episodic memory retrieval is likely to lead to long-term reward. Similarly, it should learn when episodic storage is called for. Indeed, the agent should even be able to learn what retrieval cues to set in posterior cortex in order to retrieve memories that are likely to help in deciding how to act. Put simply, TOSCA should be able to learn how to use its episodic memory most effectively in addition to learning episodic memories themselves.

4.6 Cortico-striatal circuits (intention selection and dopamine modulation)

4.6.1 Intention selection

Anatomical structure
The basal ganglia (BG) are a set of interconnected, sub-cortical nuclei which form a complex network of loops integrating cortical, thalamic and brainstem information (Alexander et al 1986). There are three main pathways from the cortex, through the BG, and back to the cortex (Figure 6). The striatum is the input nucleus of the direct pathway. It projects directly to the output nuclei of the BG, the globus pallidus interna (GPi) and substantia nigra pars reticulata (SNr). The output nuclei project back to the cortex via the thalamus, with the input returning to the same cortical module that provided the excitation to the striatum. The striatum also has a second pathway to the output nuclei, the indirect pathway. This two step inhibitory pathway provides delayed excitation to the same area of the output nuclei that the striatum inhibited via the direct pathway. The hyperdirect pathway provides a route for cortical excitation to be passed to the output nuclei of the BG.

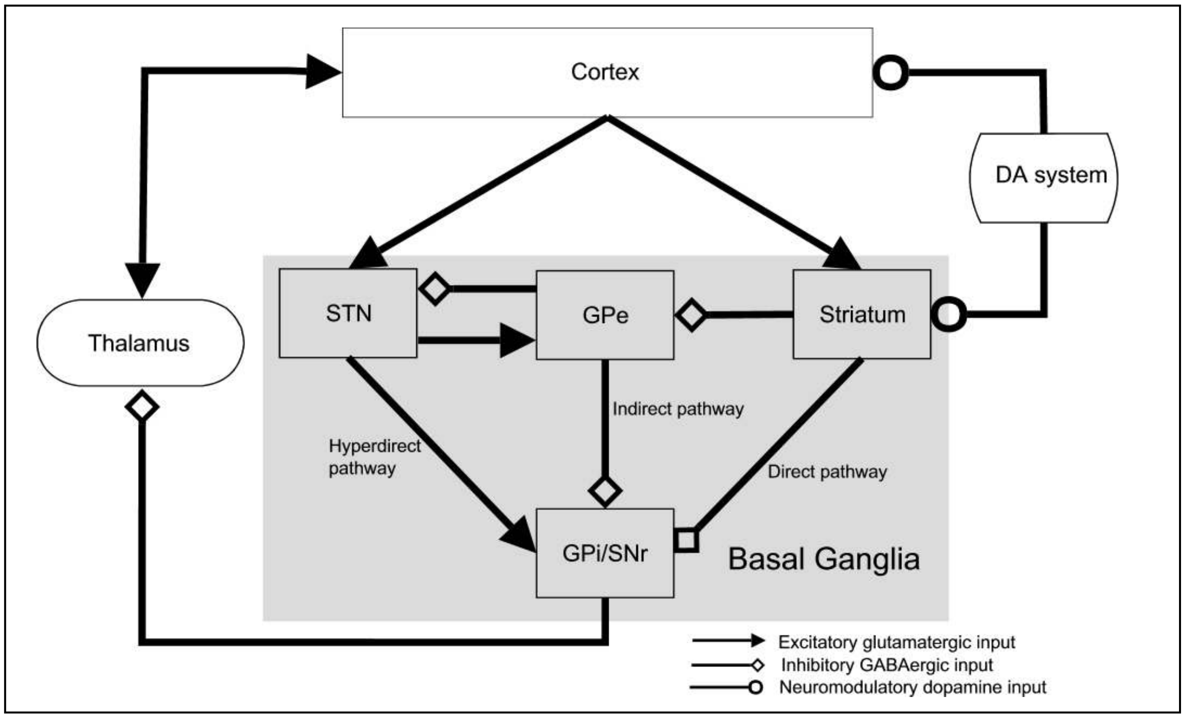


Figure 6. Schematic of a single corticostriatal loop

Each loop through the basal ganglia originates in a specific cortical area and terminates in the same area. This provides a set of parallel loops through the basal ganglia as shown by the relationship of the output channels with specific cortical areas in Figure 7. Communication between the channels occurs at the level of corticothalamic loops and cortico-cortical circuits.

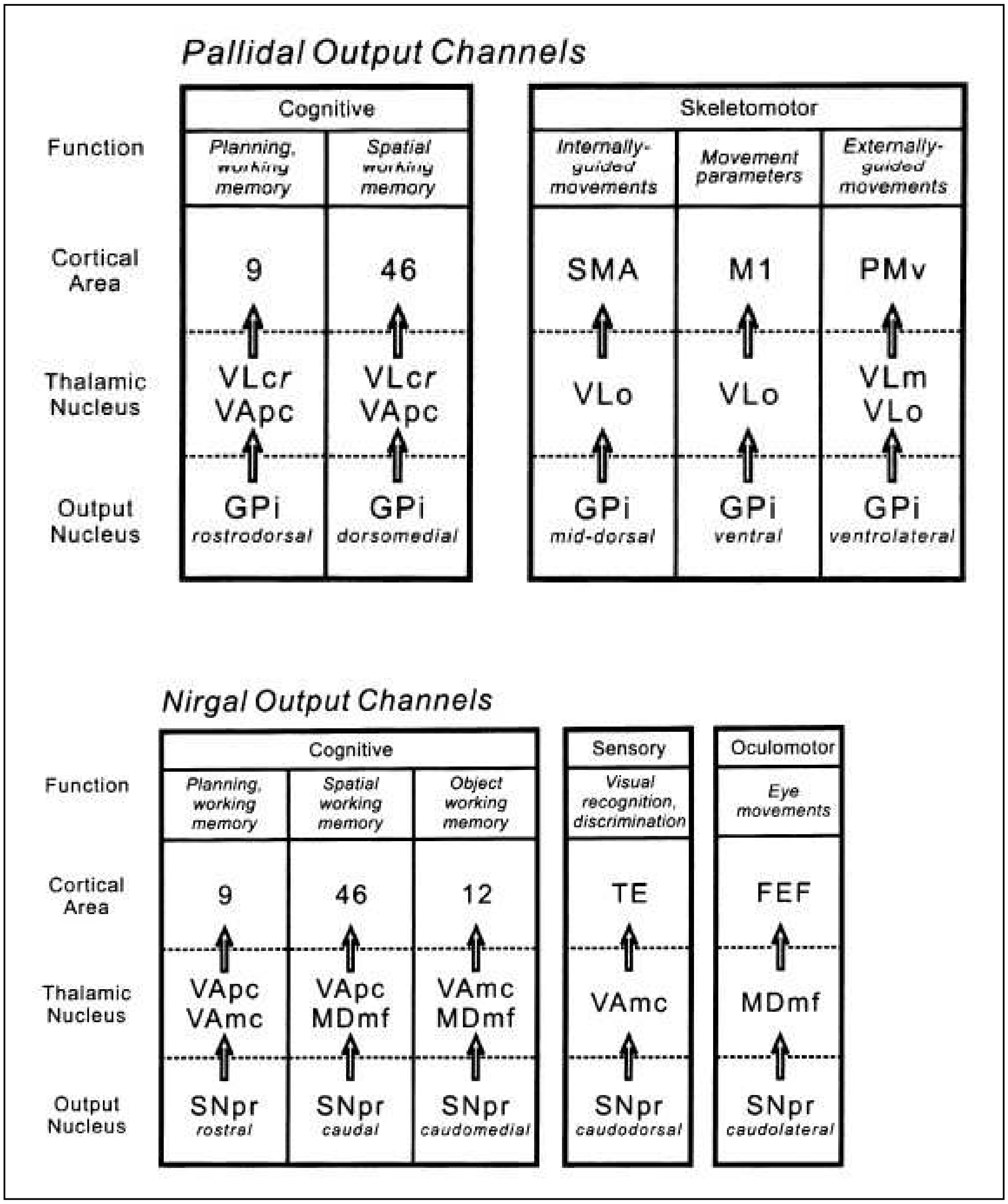


Figure 7. Basal ganglia output channels

Physiological operation

The cortical module proposes a number of contesting intentions. These are held in check by the tonic inhibitory output of the GPi/SNr acting via the thalamus. The striatum acts to decide amongst the competing intentions using information from past rewards obtained in similar environmental contexts (see section 4.6.2). The three pathways provide mechanisms for intention selection, control of the force of the release of the intention and duration of release of the intention. The presence of multiple, parallel corticostriatal loops allows for the selection of multiple intentions in parallel. Intentions that are mutually exclusive (e.g. reach for ball with left hand and scratch head with left hand) will be presented within the same corticostriatal loop, and will therefore be decided between. Intentions that can be executed in parallel (e.g. walk and talk) can be selected in parallel and thus executed simultaneously. The segregated corticostriatal loops interact at the cortical level, with the feeding of information generally from areas of more abstract intentions to more motor intentions. As an example, the first corticostriatal loop, communicating with areas in prefrontal cortex may decide that the medium term intention goal is to satisfy hunger. This decision will be passed back to the prefrontal cortex and forwarded to more motor planning areas. The next corticostriatal loop, originating from the motor planning areas, will decide that the current motor plan is to go to the cafeteria. This decision is then communicated back to the motor planning cortical area and forwarded to a shorter term motor planning area. This series of loops continues until the first action of the sequence is decided upon, perhaps rising from a chair. The medium term goal of hunger satiation remains - repeatedly selected by its corticostriatal loop when a decision between hunger and another medium term goal is required. The actions needed to fulfill that goal are executed in sequence until the goal has been met and another medium term goal attains a higher priority and is therefore selected in the corticostriatal loop.

Derived computational functionality

We assume that a central function of corticostriatal circuits is action selection (or more accurately, intention selection). Specifically, the corticostriatal circuits in TOSCA will act as a winner-take-all network to mediate between mutually exclusive intentions. The main computation is performed at the level of the striatum where the intrinsic membrane properties of the principal neurons provide the capability to differentiate between the expected reward from each of the competing intentions. When a rewarding (or aversive) event occurs, the intentions that led to the event will be strengthened (or weakened) within the striatum so that they are more (or less) likely to be selected the next time a similar environmental context is encountered.

Systems

As previously discussed, projections from posterior to anterior cortex can naturally encode associations between actions/intentions and features of the state that suggest that action.
Multiple different, and potentially, conflicting intentions can be activated in parallel and it will often be necessary to select among conflicting actions. The neuroanatomy of corticostriatal circuits make them particularly well-suited to this function and interactions between cortex and basal ganglia will be crucial in doing so. Interactions between this system and the dopamine system will also be crucial for learning in TOSCA. Specifically, when an action leads to unexpected reward, the value of that action in the current state/context will be increased (see section 4.7.2) by potentiating the cortical associations between the state features and the action representation. The corticostriatal action-selection system will be sensitive to these values, so that when that action is proposed in similar states in the future, its probability of being selected will be higher.

4.6.2 Modulation of action contingencies via dopamine

Anatomical structure

Dopamine producing neurons are located in two midbrain nuclei, the ventral tegmental area (VTA) and the substantia nigra pars compacta (SNc) shown in Figure 8. They receive excitatory input primarily from the pedunculopontine tegmental nucleus (PPTN), which conveys information about the occurrence of primarily rewarding events, and prefrontal cortex and inhibitory input from the ventral striatum. They project to the prefrontal cortex and striatum where they fire in a phasic fashion to release dopamine in response to rewarding situations (Romo & Schultz 1990, Schultz 1996).

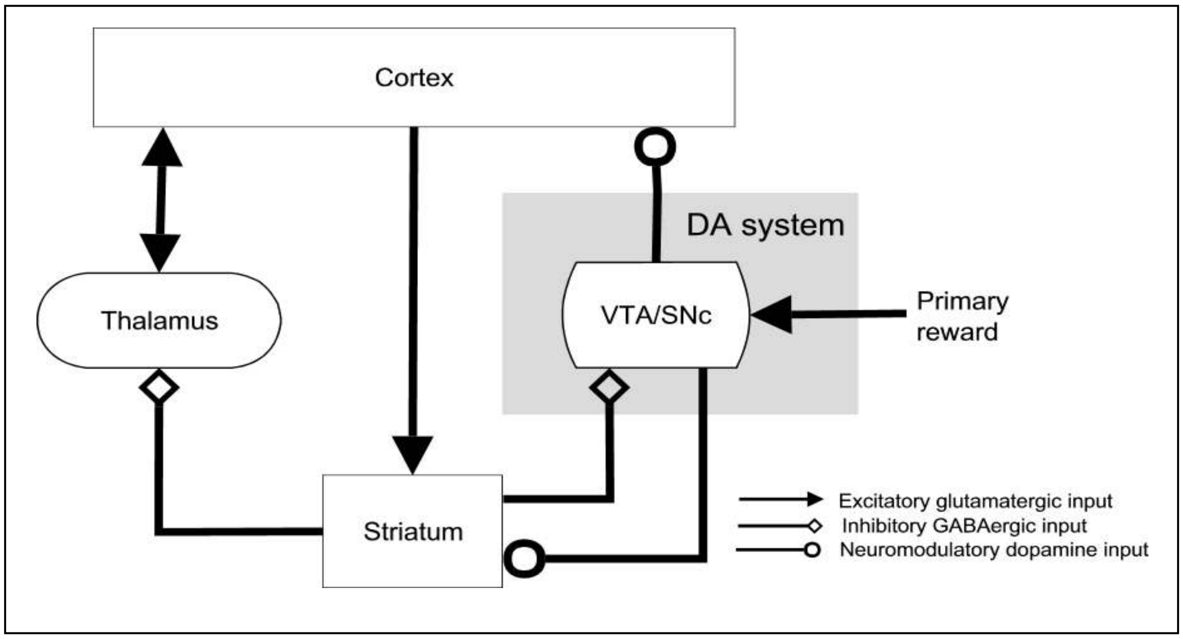

Cortex
DA system
Thalamus
VTA/SNc
Primary reward
Striatum
Excitatory glutamatergic input
Inhibitory GABAergic input
Neuromodulatory dopamine input

Figure 8. Schematic of main connections of the dopaminergic system

Physiological operation

The corticostriatal loops of the basal ganglia are the substrate for selecting between intentions (section 4.7.1). Learning of the correct intention in a given environmental context is under the control of the dopaminergic system. When a reward is encountered, the synaptic strengths in the corticostriatal circuits that were activated prior to the reward are increased. This makes it more likely that the same intention will be executed in a similar environmental context on future occasions. An unexpected (primary) reward elicits a phasic response in the dopaminergic neurons of the VTA/SNc. When a conditioned stimulus (CS) has been learned to reliably predict an upcoming reward, the time of response of the dopamine neurons shifts to coincide with the CS. These phasic releases of dopamine are utilized in the recipient structures to direct learning. The action of phasic dopamine signals is to increase synaptic strength by a 3-factor learning rule. In this rule the relative timing of synaptic input, neuronal firing and dopamine pulse conspire to dictate the amount of learning from a single rewarding event.

Derived computational functionality

Dopamine neurons have long been associated with reward learning and rewarded behavior, partly because of clear evidence of their key role in drugs of addiction (DiChiara, 1999), and because they are among the best targets for self-stimulation. The observation that the activity of dopamine cells in the monkey midbrain in reward-learning tasks closely follows the form of a key training signal in reinforcement learning (the temporal difference prediction error), is an important backdrop for TOSCA. In particular, temporal difference based RL methods will serve to modulate state-action associations by potentiating associations between clusters in posterior cortex (representing complex internal state information) and clusters in anterior cortex (representing internal and external action intentions).

Systems

The dopamine system is tightly bound to the corticostriatal system, mediating learning in the prefrontal cortex and both divisions of the striatum. This system is also now known to provide neuromodulatory input to the hippocampal and thalamic systems.

4.7 Dopamine reward circuits [Intrinsic Reward and its Neural Basis]

Anatomical structure

Recent studies (Kakade & Dayan 2002, Dayan & Balleine 2002) have focused on the idea that dopamine not only plays a critical role in the extrinsic motivational control of behaviors aimed at harvesting explicit rewards, but also in the intrinsic motivational control of behaviors associated with novelty and exploration. For instance, salient, novel sensory stimuli inspire the same sort of phasic activity of dopamine cells as novel rewards (Schultz 1998, Horvitz etal. 1997}. However, this activation extinguishes more or less quickly as the stimuli become familiar. This may underlie the fact that novelty itself has rewarding characteristics (Montague etal.1996).

The novelty-based release of dopamine onto one of its major targets, the striatum, causes both general psychomotor activation (Hooks & Kalivas 1994) and also specific exploratory or seeking behaviors such as approach that cause animals to engage with those novel stimuli. Approach of this sort is a Pavlovian response - it is like a pre-wired action inspired by novelty (and also reward prediction). Theoretical treatments (Kakade & Dayan 2001, Kakade & Dayan 2002) have directly related the dopamine activity with mechanisms for controlling exploration in the RL literature such as exploration and shaping bonuses (Sutton, 1993, Dayan & Sejnowski 1996, Ng et al. 1999) effectively completing the circle of interaction between computational, psychological and neural approaches. In TOSCA, we will explore a wider set of mechanisms by which animals control and benefit from exploration, using it to build sophisticated mechanisms for manipulating and exploiting novel environments. This wider set of mechanisms include the desire for mastery over one's environment and often leads to purposeful and sustained experimentation, as well as the motivation of an agent in a social setting to be liked by other agents (like-me) which leads to imitative behavior in social settings.

Various studies have also considered the neural basis of the assessment of novelty. Of particular relevance are two further neuromodulators, acetylcholine (ACh) and norepinephrine (NE), which are known to be involved in uncertainty and unexpectedness, and also to interact with the dopamine system. Theoretical treatments of these (Dayan &Yu 2003,Yu & Dayan 2002 focus on their roles in reporting specific sorts of uncertainty---uncertainty arising from ignorance (which is what should drive exploration) and uncertainty arising from environmental stochasticity (which should not). The difference between these forms of uncertainty is relative to models of the environment, which form a key component of any theory of novelty. The ideas on ACh and NE are in their infancy; there is scope for a productive interaction between our explorations via TOSCA and future experiments and theory on the drives and effects of NE and ACh.

Derived computational functionality

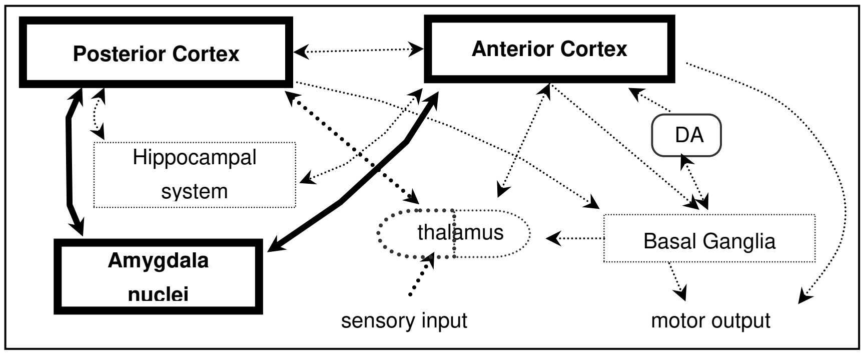
The intrinsic motivations listed above will serve as mechanisms for providing internal reward to the agent and this in turn will help direct the agent's behavior during exploration and play both in the presence and absence of externally specified tasks. These internal rewards will lead to the learning of useful mental and physical skills in the form of *options* or abstract actions that in turn will become available to the reinforcement learning system in TOSCA as actions. This will allow an incremental buildup of a hierarchy of useful cognitive and physical skills by the agent that would not be possible in the absence of intrinsic motivations.

Systems

The dopamine system is tightly bound to the corticostriatal system, mediating learning in the prefrontal cortex and both divisions of the striatum. This system is also now known to provide neuromodulatory input to the hippocampal and thalamic systems.

4.8 Cortico-amygdala circuits (emotion, state-dependent storage & retrieval)

Anatomical structure



The amygdala formation is composed of multiple subparts typically grouped into the Medial group, the Central group, and the Basolateral group. The latter, forming the baso-lateral amygdala (BLA) is an evolutionarily recent structure in contrast to the central and medial amygdala which are phylogenetically ancient. Central and medial amygdala nuclei are strongly connected to brainstem and hypothalamic structures and are implicated in visceral and hormonal modulation. See Figure 9 (left).

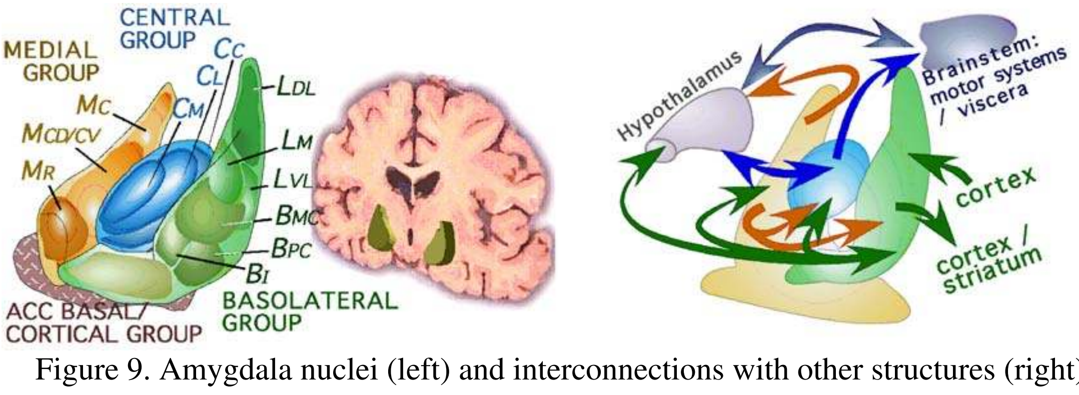


Figure 9. Amygdala nuclei (left) and interconnections with other structures (right)

The BLA is highly connected with portions of cortex including medial and lateral prefrontal cortex, sensory association cortex, as well as ventromedial frontal, rostral insular and rostral temporal cortical areas, and the medial thalamus and ventromedial basal ganglia. (See Figure 9 (right)). Connections from amygdala to cortex have recently been confirmed to preserve topographic organization (Amaral et al., 2003; Alheid 2003; Price 2003).

Physiological operation

Basolateral amygdala's topographic connectivity with anterior cortical regions is capable of "priming" or activating at a subthreshold level region in anterior cingulate cortex, prefrontal cortex, and orbitofrontal cortex among others. Reciprocal activation is in evidence; i.e., amygdala and cortex activate each other (constrained by their topographic projection patterns).

Derived computational functionality

Analyses currently in progress suggest computational utility of cortico-amygdala circuitry; these include the reciprocal physiological priming effect described above, learning in both cortex and amygdala, and interaction between amygdala and hypothalamus. This work is currently being written up for publication (Hearn and Granger, in prep) and it is anticipated that the corresponding analyses will be included in the TOSCA architecture.

Systems

Cortico-amygdala circuits are integrated with cortico-striatal circuitry; these circuits have effects on behavior from sensory recognition to motor function, as well as on episodic memory storage and retrieval.

4.9 Cerebellum

Anatomical structure

The cerebellum can be subdivided into the cerebellar cortex and the deep cerebellar nuclei, which sit on top of the cerebellar peduncle (Figure 10). The largest subdivision of the cerebellar cortex in humans is the cerebrocerebellum which occupies most of the lateral cerebellar hemispheres and receives input from many areas of the cerebral cortex. The phylogenetically oldest part of the cerebellar cortex is the vestibulocerebellum, which comprises the caudal lobes. The third division is the spinocerebellum, which occupies the median and paramedian zone of the cerebellar hemispheres. The deep cerebellar nuclei are embedded within the white matter of the cerebellum. The connections between the cerebellum and other parts of the nervous system occur by way of three large pathways called the cerebellar peduncles. The middle cerebellar peduncle is an afferent pathway arising mainly in the pons and the superior cerebellar peduncle is an efferent pathway from the deep cerebellar nuclei to the thalamus.

The majority of cerebral cortical inputs to the cerebellum arise in the primary motor and premotor cortices of the frontal lobe, the primary and secondary somatic sensory cortices of the anterior parietal lobe and the secondary visual regions of the posterior parietal lobe. The cerebellum projects mainly to the upper motor neurons in the cerebral cortex via relay neurons in the thalamus.

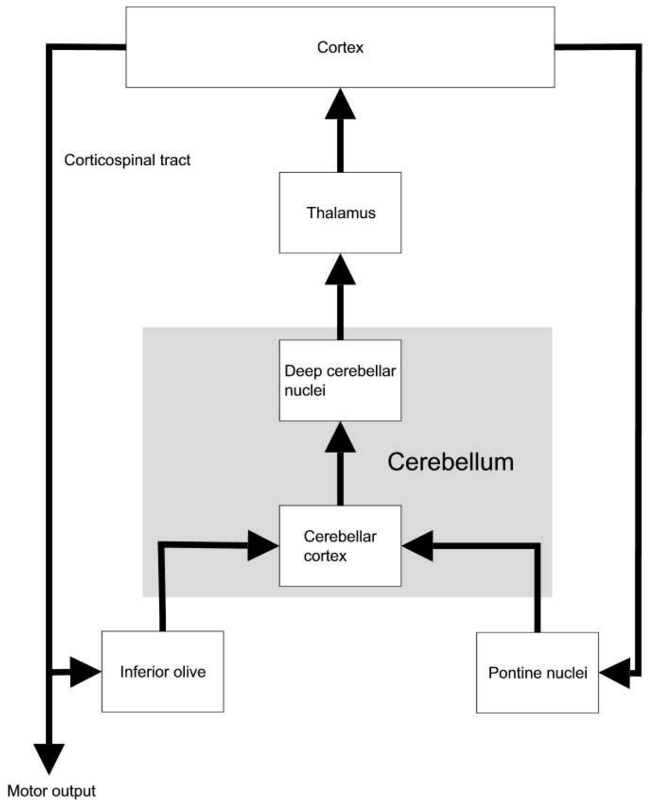


Figure 10. Schematic of the major connections of the cerebellum

Physiological operation

The cerebellum influences movements by modifying the activity patterns of the upper motor neurons. The primary function of the cerebellum is to detect the difference (or motor error) between an intended movement and the actual movement and, through its projections to the upper motor neurons, to reduce the error (Gluck et al 2001). These corrections can be made both during the course of a movement and as a form of motor learning when the correction is stored.

From lesion studies it has been found that the cerebellar loop is critical for the performance of planned, voluntary, multijoint movements. The activity of the cerebellum instructs the motor cortex in the direction, timing and force. For ballistic movements these instructions are based entirely on predictions about their outcome.

Derived computational functionality

In the TOSCA architecture the cerebellum acts to store complex motor programs as they are learned. Individual movements, originally used as separate parts of a complex movement sequence, will be gradually compiled into motor programs in the cerebellum. These motor programs generate the appropriate motor sequences on demand and through supervised learning gradually make execution of the movement sequences smoother and better coordinated (Gluck et al 1994).

Systems

The cerebellum interacts primarily with the cerebral cortex. In early phases of motor learning, the motor programs will be simple and proposed by the cerebral cortex. When the intended action has been selected by the basal ganglia, the action will be executed by the primary motor cortex.

The cerebellum will receive information about the intended outcome of the action from the motor cortices and the outcome of execution of the action from the sensory cortices. The difference between the intention and the outcome will be used by the cerebellum for learning of the motor action. The next time the same motor action is proposed, the cerebellum will have an influence on the execution of the action and will use the error in the action execution to continue learning.

5. Key emergent functional properties: Representation & Control

The previous section lays out our vision for TOSCA at the level of brain systems and circuits. It explores the physiology we are trying to capture in TOSCA as well as the low-level computation being performed in individual brain systems and in brain circuits. However, it is down at a level where it is often difficult to see how human-level behavior emerges from these components and their connections.

Two primary features of the design of TOSCA are its representational system and its control system. The next two sections present initial overall views of how the underlying mechanisms presented in section 4 operate together to generate an overall control mechanism to engage its environment, and to learn rich representations about that environment. Learning permeates the operation of the TOSCA system: the system is continually learning and cannot help but learn, and thereby builds up representations from combinations of perception and prior knowledge, as well as building up control knowledge.

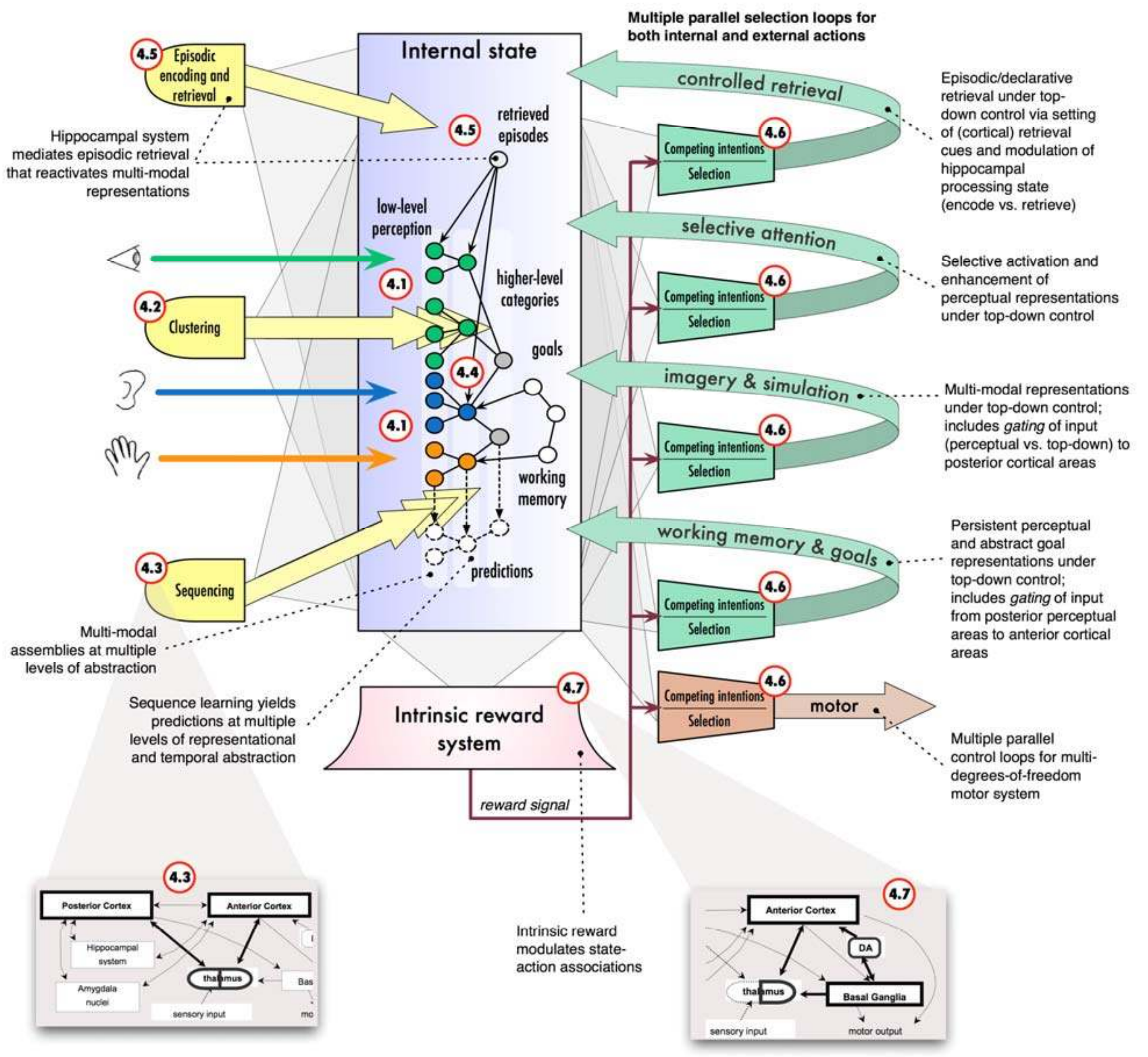

Multiple parallel selection loops for both internal and external actions
4.5 Episodic encoding and retrieval
Hippocampal system mediates episodic retrieval that reactivates multi-modal representations
Internal state
4.5 retrieved episodes
low-level perception
higher-level categories
goals
working memory
predictions
4.1
4.2 Clustering
4.4
4.3 Sequencing
Multi-modal assemblies at multiple levels of abstraction
Sequence learning yields predictions at multiple levels of representational and temporal abstraction
4.7 Intrinsic reward system
reward signal
Intrinsic reward modulates state-action associations
controlled retrieval
selective attention
imagery & simulation
working memory & goals
4.6 Competing intentions
Selection
motor
Episodic/declarative retrieval under top-down control via setting of (cortical) retrieval cues and modulation of hippocampal processing state (encode vs. retrieve)
Selective activation and enhancement of perceptual representations under top-down control
Multi-modal representations under top-down control; includes gating of input (perceptual vs. top-down) to posterior cortical areas
Persistent perceptual and abstract goal representations under top-down control; includes gating of input from posterior perceptual areas to anterior cortical areas
Multiple parallel control loops for multi-degrees-of-freedom motor system
Posterior Cortex
Anterior Cortex
Hippocampal system
Amygdala nuclei
thalamus
sensory input
DA
Basal Ganglia
motor output

Figure 11. Depiction of functional organization of TOSCA

Figure 11 depicts how the neural systems described throughout section 4 are functionally organized to produce useful behavior. The circled numbers correspond to subsections in subsection 4 where the underlying circuits that support the functionality are described. As can be seen in the bottom of the figure in the expanded diagrams, a given *functional* module in the figure (such as sequencing: 4.3, or intrinsic reward: 4.7) are realized by circuits that involve multiple brain systems.

The figure is organized on the left with the major contributors to the active internal state, which itself is organized vertically in the middle of the figure. Perception from various modalities (vision, auditory and touch are shown) provide low-level input [4.1]. Clustering [4.2] and sequencing [4.3] work from low-level perception, as well as other available internal state structures, to create more abstract structures that in turn become available for further clustering and sequencing as well as input for control (on the right side of the diagram) [4.6, 4.7]. The control system [4.6] consists of many parallel loops that work across both external actions (the motor system) and internal "mental" operations, including goal selection, persistence of working memory and goal structures, internal simulation, selective attention, and retrieval from episodic memory [4.5]. The reward system sends signals to the action selection modules (4.6) to tune performance through reinforcement learning.

5.1 Representation and memories: Multi-modal sequences of clusters and self-organizing maps

General properties of memories

The deceptively simple operators of sequences and categories, and the resulting data structures (hierarchically nested sequences of categories), interacting with special-purpose structures such as hippocampal, amygdala, and striatal formations give rise to the complete set of internal "knowledge representations" that occur in the TOSCA architecture. This surprising finding is in a way at the core of the TOSCA effort: it is a discovery of how advanced complex behaviors can be constructed from apparently simple interacting components.

Of particular importance is the emergence of interactions in what can be termed the primary architectural loop in TOSCA: the cortico-striatal loop. This set of circuits accounts for the vast majority of all the "real estate" in the entire system. Its behavior can be succinctly summarized thus: the representations generated by cortico-cortical systems can function as internal representations or models of states, which in turn can be tested and adaptively modified via reinforcement learning in cortico-striatal loop interactions. The resulting "adaptive exploitation" enables the construction of large and elaborate internal representations, and fits between those representations and the environment, via these basic powerful brain circuit mechanisms.

A note on learning via long-term potentiation (LTP)

In TOSCA, memories are stored via synaptic LTP, which operates via a set of well-worked out and extensively published and replicated rules and mechanisms. Many of these are unfamiliar to the field of psychology, and yield unusual memory effects in the architecture, all consistent with observed psychological phenomena.

Temporary memory:

Initial storage makes initial changes to synaptic weights. (Initial memory)

Consolidation:

If no new signals address those same synapses (storage sites) within the next 15-30 minutes (the synaptic consolidation period), those changes become permanent, i.e., irreversible. (Permanent memory).

Reversal:

If interfering signals do arrive at these storage sites within the synaptic consolidation period, the weight changes can be reversed. This can result in entirely forgetting or just "shaping" / altering the stored memory.

Elaboration:

Once stored, the memories can be internally accessed (via cortico-hippocampal loops) and give rise to internal practice and elaboration. In particular, new memories can be created to elaborate the initial memories, linking the initial memories to additional related memory items (e.g., seeing one car can become related to having driven in either that car or another car, etc), both enriching and altering the memories. This occurs via new synaptic recruitment and storage at new additional sites.

Thus all memories in the proposed architecture begin as temporary memories and can either become permanent (and possibly elaborated) or reversed (erased). Memories are stored where they are sensed or acted. There are no separate "locations" for memories of different durations.

Emergence of (multi-modal) categories = internal grammars

Two features of brain circuits past the sensory periphery are notable:

i) circuits for different modalities (e.g., vision, audition) are remarkably similar (though not always identical; some of the gradient differences will be discussed separately); and
ii) the majority of circuits receive inputs from multiple modalities. Thus communication among cortical regions consists of a single, shared, cross-modal internal representation language, regardless of the particular information being conveyed.

Individual cortical regions compute clusters (i.e., similarity-based categories) and sequences (chaining), via different components of their intrinsic circuitry. These two components, interact to produce sequences of categories (see Rodriguez et al. 2004). The output of one thalamocortical circuit is input to others with identical or near-identical structure; these thus produce sequences of categories of sequences of categories …, effectively nesting the product of one "level" of processing into downstream processing products.

Successive nesting creates increasingly deep hierarchical "trees" of sequences of clusters. (Feedback from downstream to upstream regions participates actively in this process; partial activation of a downstream region has the consequence of increasing the probability of response of its potential upstream input constituents, acting in effect like "expectations" that those inputs will occur.)

These cortical mechanisms interact with hippocampal time dilation and contraction, amygdala "toggling" of salient features, and striatal reinforcement learning in cases of relevant feedback. Together the system produces incrementally constructed and selectively reinforced hierarchical representations consisting of nested sequences of categories (Granger 2006).

Figure 12 is an abstract illustration of successive stages of a representation so constructed. Initial simple input features (e.g., visual spots or edges; auditory frequencies or formants) transduced by front end mechanisms are learned by earliest, specialized stages (denoted in the figure by single letters A, B, etc). Their encoded outputs are input to downstream structures which learn clusters (categories of similar inputs) and sequences of clusters; further downstream regions learn sequences of clusters of sequences of clusters, and so on.

Each downstream region, depending on its pattern of connectivity with its inputs, may exhibit a "bias", preferring inputs with particular characteristics; these are genetically programmed and little is yet known of their layout, though work in quantitative neuro-anatomy is advancing knowledge in this realm. In TOSCA we will assume the existence of such biases, which cause different cortical regions to become increasingly "specialized" via learning for the particular feature combinations that they are most likely to successfully "compete" via lateral inhibition.

In practice, it would be prohibitively expensive computationally to learn all such combinations of features, but combinatorial explosion is avoided by two primary mechanisms:

i) Bias: Of all the possible combinations of features that could occur, only some actually do, and, as just mentioned, some combinations are preferred over others;
ii) Competition: With learning, oft-traversed regions become increasingly strengthened and, via lateral inhibition of neighboring regions, become what may be thought of as "specialists" in certain types of inputs, competing to respond.

Due to the described architectural arrangement, early upstream areas tend to respond to generic features and simple feature assemblies, but downstream regions respond with increasing selectivity to only specific assemblies, typically those that occur as patterns within oft-seen stimuli.

As a concomitant, further downstream regions should be expected to selectively respond to larger or longer patterns, both in visual and auditory domains. As most visual inputs consist simply of different arrangements of the same sets of primitive input features, it is expected that patterns of brain activation should be extremely similar in response to many different visual inputs, but that the similarity of those brain activation patterns ought to correspond to the similarity of their inputs, that is, activation patterns ought to be more similar for similar inputs, and more different for different inputs.

Moreover, if cortical regions are competing to respond to a given input, they should exhibit "category boundaries," that is, the responses to images within a category (e.g., faces versus houses) should be more similar to each other than the images themselves are. Put differently, even highly different faces are likely to generate very similar cortical response patterns, whereas the similarity between any face and any house (as long as it is not a house that looks like a face!) should be more different than any two faces or any two houses.

These three sets of predictions from the model (distributed representations, similarity of patterns, and category boundaries) turn out to be controversial: depending on the analysis methods, neuroimaging studies have been used to support a number of still-conflicting hypotheses. TOSCA's architectural design, as described, is concordant with some of the most prominent findings, in which distributed, overlapping patterns occur in response to images of, say, faces vs. houses; more similar inputs tend to generate more similar responses; and responses to images within perceptual categories are more similar than responses to images across categories (Haxby et al. 2001; Pietrini et al. 2004; Furey et al. 2006; Hanson et al. 2004).

The representations thus far described bear some resemblance to long-term semantic memory: they are

- permanently stored,
- contain (learned) relations among components,
- tend to refer to categories and abstracts rather than individuals, and

- lend themselves to representations of generic types, e.g., “letters,” “speech sounds,” “jeeps.”

Characteristics of grammars (sequences of categories)

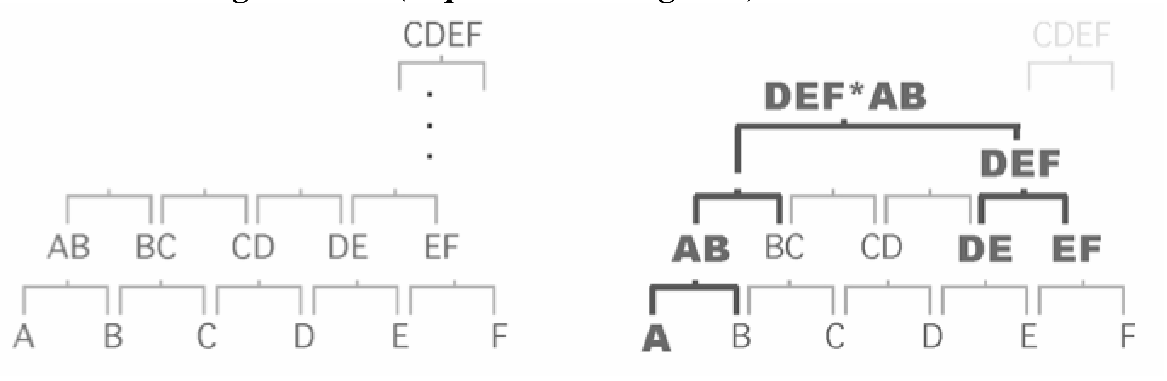


Figure 12. Illustration of hierarchies constructed by telencephalic architecture. Initial features generate successively nested sequences of categories of features (left). Additional exposure eventually (right) selectively strengthens sequences that recur (e.g., AB), weakens those that do not (e.g., CDEF), and constructs new sequences of categories as they occur and recur (e.g., DEF followed by a category that may include any of A-F (denoted here by a *) followed by AB).

The emergent data structure of the telencephalic system, statistically learned nested sequences of categories (as illustrated in Figure 12) is a superset of the structures that constitute formal grammars. The nested sequences of clusters are equivalent to ordered sequences of “proto-grammatical” elements such that each element represents either a category (in this case a cluster) or expands to another such element (nesting), just as grammatical rewrite rules establish new relations among grammatical elements.

Learning of the model’s representations as thus far defined (nested sequences of categories) constructs one type of semantic network referring to categories of objects, including relations among their internal parts (e.g., the hood, windows and trunk of a car).

Still to be specified are representations of a kind often occurring in such specifications – e.g., abstractions of relations (“in front of,” “above,” “containing”). These arise in a way compatible with hypotheses of “simulations” (Barsalou et al., 1999), i.e., learning specific instances in which objects are in the relation (a plate above a table, a hand above a paper, a window above a desk) generates not just representations of and among the particulars, but also abstract hierarchical representations of the relations themselves, which in turn become applicable to new inputs (a plane above a mountain) not previously seen.

The system extracts feature subsets as it learns, and generates regional cortical “specialists” as described earlier. Physical arrangements of objects in which one is higher in the visual field than the other, for instance, lead to (relatively early) specialists that characterize the relationship between them. This relationship comes to have the verbal associations “above” and “below” (among others), and these relations come to internally define the corresponding abstract relations. This is also an area of still-active study in the architecture.

Emergence of high-level cognitive representations

The incremental nature of the "nested sequences of categories" data structure enables it to grow in function, simply by adding new copies of telencephalic thalamo-cortico-striatal-limbic loops – this functional growth corresponds to the incremental addition of "rules" acquired by the grammar. As more telencephalic "real estate" is added, the data structures that are constructed correspond to both longer and more abstract sequences, due to iterative nesting. Even regions of telencephalon with identical (or nearly identical) computational function nonetheless receive inputs from different sources, thus changing the feature combinations on which they operate (but see Galuske et al. 2000; Preuss 1995; 2000).

Proceeding "downstream" through the architecture, the outputs of one area are input to the next area. Successively more complex data structures should emerge, capturing increasingly complex representational concepts. Thus differential branching pathways through the architecture come to "specialize" in different functional realms.

Topics of ongoing study in the architecture concern the emergence of representational abstractions much-studied in psychology and in artificial intelligence, such as type-token distinctions (e.g., between "car" and "this car"), which enable distinguishing between individuals and categories. Initial study indicates that cortico-hippocampal interaction plays a role in this process, enabling the generation of different specifiers, qualifying cortico-cortical representations.

As mentioned, of the large set of all possible assemblies of features, only a small subset seem to be readily learned by biological organisms; there apparently exist species-specific biases that shape animals' (including humans') interpretations of various inputs. For instance, in response to very little data, humans will interpret certain coherent point-source motions as biological motion (e.g., when lights are affixed to the limbs of people moving in an otherwise dark environment); will interpret many distorted inputs as face-like; will interpret many sounds as speech-like, and so on. It is assumed that these biases may arise from developmental pre-selection (via mechanisms to be discussed elsewhere) of some cortico-cortical pathways that will selectively respond to particular types of feature assemblies.

It should be emphasized that all of these growing representational traits are hypothesized to arise directly from the hierarchical sequences of categories representations as manipulated by cortico-striatal loops. From low-level sensory beginnings, the abstractions grow to encompass the apparently full range of high-level cognitive concepts. This will be a central topic of study in the architecture.

Emergence of language

It has already been seen that the primary internal representation, hierarchically nested sequences of categories, is a form of grammar, i.e., shares the characteristics of formal grammar systems, though as described it has been used thus far for representations of sensory and motor sequences, not typically associated with grammars. In the TOSCA architecture, then, all internal representations are couched in the formalism of grammars (of this specific type).

If this representational hierarchy grows large enough (a function solely of the space of cortico-cortical structure allocated), the resulting abstractions become symbolic descriptors. At this point, the already grammar-based representation becomes the internal basis for linguistic representation. In other words, in the TOSCA architecture, grammars do not arise abruptly in service of linguistic abilities – rather, grammars are present throughout, and language arises as the representational hierarchy grows sufficiently large.

As described earlier, far-downstream areas are assumed to come to identify increasingly abstract symbolic descriptors (see Tables 1, 2) that are statistically repeated in relevant situations. These include definitions of words as well as the abstract relations that underlie the words' meanings.

Figure 13 illustrates structures occurring in response to simple sentences ("John hit Sam") as input. Construction of sequences (e.g., S11, "John" followed by "hit"), and categories (e.g., C21, "hit" and "kissed," items that can follow "John") are combined in successive downstream regions (n+1, n+2, etc.) to create "proto-grammatical fragments" corresponding to internal representations of linguistic structure information.

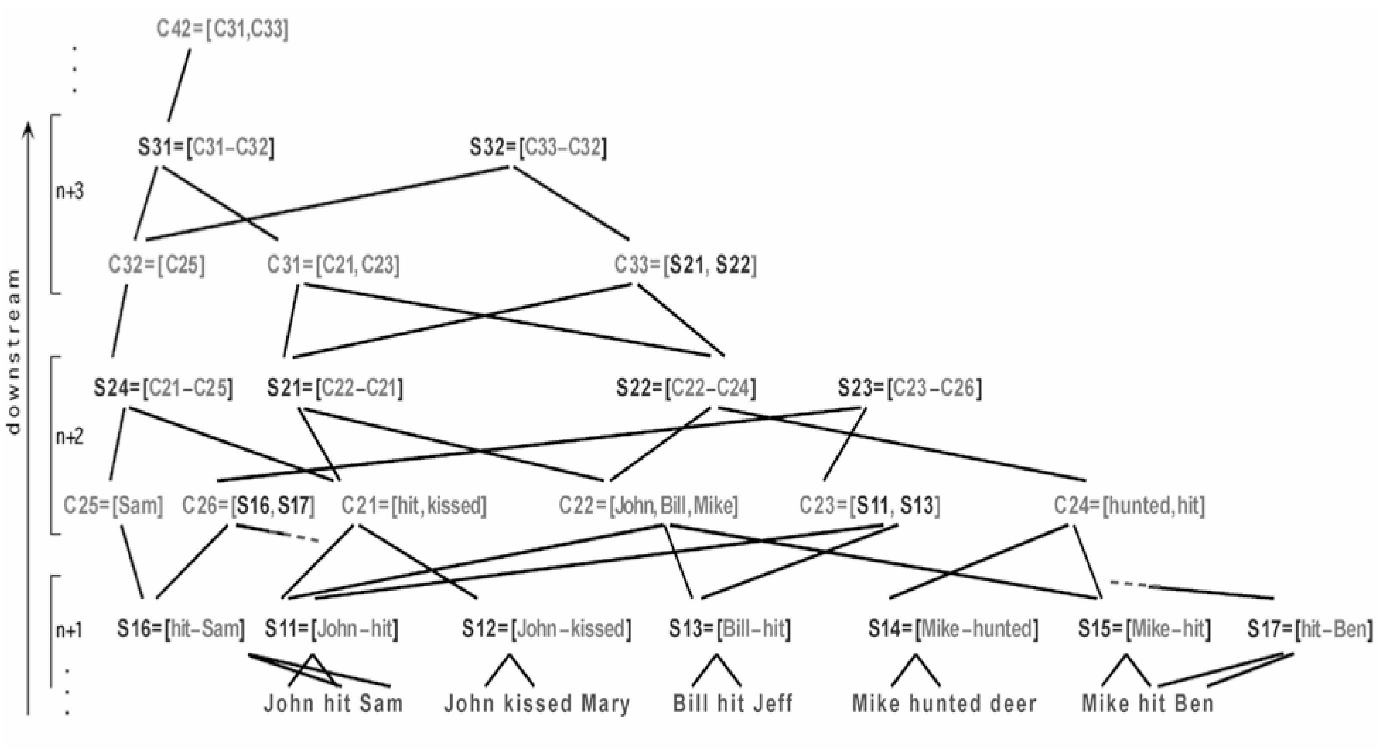


Figure 13. Nested sequences of clusters as sample proto-grammatical fragments educed from input strings

It is worth noting that the generated structures can be used both a) in the processing of subsequent novel inputs and b) in the generation of arbitrary new strings that will conform to the rules inherent in the learned internal representational structures.

The resulting "generative" nature of the representations is worth emphasizing, addressing a crucial aspect of linguistic grammars that can otherwise be absent from some purely input-processing or parsing mechanisms. A potentially infinite set of strings can be generated from the internal sequences of clusters, and the strings will be consistent with the internal grammar (see, e.g., Pinker 1999; Hauser et al. 2002; Fitch & Hauser 2004; Pinker & Jackendoff 2005).

It is also noteworthy that the grammar does not take the form typically adopted in attempts to formally characterize the syntactic structure of natural languages (such as English). The protogrammatical fragments capture regularities that are empirically seen to suffice for both parsing and generation, and have the structure to account for rule-like behaviors that characterize linguistic behavior. Research is currently in progress to study the formal relations between typical linguistic grammars, and protogrammatical fragments that are emergent from nested sequences of clusters.

(An additional characteristic of language that challenges researchers is the seeming effortlessness with which children learn language – readily contrasted even with the comparatively laborious training typically required for adults learning a second language. It is hypothesized that an innate bias related to sequences of categories of vocal utterances (speech) may lead (in larger-brained organisms) to a downstream bias for certain sequences of categories of assemblies of speech sounds (words). This may at least in part account for this much-studied but still elusive nature of innate language capacity; see Granger 2006).

5.2 Control: External motor resources and internal cognitive resources

In the previous section we described the emergent nature of representation in TOSCA—how both the present and the past come to be represented as multi-modal sequences of clusters at multiple levels of abstraction. In this section we describe how TOSCA exploits those representations to achieve adaptive, moment-by-moment control of both external motor effectors and internal cognitive states. The TOSCA theory of control is based on parallel loops of action selection contingent upon the representations described above and continuously modified by the intrinsic reward system. In the remainder of this section we first describe this general theory of control, followed by discussion of some of the specific control loops devoted to motor control and cognitive control.

5.2.1 General properties of control

TOSCA's general theory of control is based on three fundamental principles that have considerable biological and functional motivation: (1) fine-grained, parallel selection loops for both external and internal actions; (2) action selection potentially contingent upon multiple aspects of the internally represented state; (3) reinforcement learning of control realized by a rich intrinsic reward system. We now briefly summarize each of these in turn:

- *Fine-grained, parallel selection loops for both external and internal actions.* As we described above (Section 4.6) we assume that action selection is mediated by cortico-striatal loops, and more specifically, that multiple regions of frontal cortex represent competing intentions for action. There is growing neuroanatomical evidence that these loops are quite segregated, so that the frontal-striatal system is best understood as consisting of many fine-grained selection loops operating in parallel. Functionally, this organization is well-suited to support the real-time control of a motor system with many degrees of freedom (and much of the evidence for the segregation of frontal-striatal loops comes from detailed studies of mammalian motor systems), but as we outline in more detail below, it also naturally extends to the control of a cognitive system with multiple independent resources. Furthermore, in each specific case of cognitive control that we propose below, this extension is consistent with the existing biological evidence. Figure

11 above (right half) summarizes five major classes of control loops that we intend to model in TOSCA; each major control loop may be further broken down into separate finer-grained loops as described below.

- *Action selection contingent upon multiple aspects of the internal state.* One of the hallmarks of human cognition is its ability to adaptively exhibit arbitrary and novel behavioral contingencies. In prominent symbolic computational models of human cognition such as ACT-R and Soar, production rules play an important role in supporting this flexibility. More specifically, a critical property of production rules is that they allow action selection to be contingent upon any arbitrary features of the internally represented state (via the patterns in the "condition" side of the rules). Such functionality is directly supported in TOSCA by the massive inputs into frontal cortex from multiple posterior and anterior brain regions; this connectivity pattern and the neuro-anatomical evidence for it was described in Section 4.6. Figure 11 above (right half) depicts this broad contingency: each of the control loops starts with the activation of a set of *potential actions* that may be triggered by any aspect(s) of the internal state. These potential actions (or *intentions)* are represented in specific distinct frontal regions that participate in the segregated action loops described above.

- *Reinforcement learning of control.* Although the learning of behavioral and cognitive control in TOSCA ultimately depends on the interaction of multiple learning mechanisms in the architecture (including episodic encoding, clustering, and sequencing), the *direct* basis for learning control is reinforcement learning (RL) as realized by TOSCA's intrinsic reward system. We believe that much of the power of our proposed architecture will derive from the interaction of RL with the representational capacities of the system described above in Section 5.1. The functional neuroanatomy of this reward system was described in Section 4.7; the parallels to abstract properties of algorithms for reinforcement learning are well known (Shultz et al. 1997). The specific properties of RL in TOSCA are as follows:

- *Intrinsic reward.* All reward in the system is internally generated. "External reward" is translated from a sensation into a form of internal reward – there is no direct line from the environment to a reward signal. The internal rewards include intrinsic motivations or drives such as: *novelty, mastery,* and *exploration*. (See Section 4.7.2 above for the neural bases for these drives). These drives interact to determine the nature of both exploratory and task-driven behavior. The computational implications of this system for the nature of exploratory learning are significant and we draw them out in more detail in the next section below.

- *Exploitation of generalizations admitted by cortical representations.* In the previous section we described two critical ways that generalization is supported by cortical representations: via the emergence of *hierarchical clustering* realized by thalamocortical loops, and via the self-organization of cortex into *maps* that support immediate similarity-based generalization. TOSCA's RL system will adaptively exploit these generalizations: contingencies (more specifically, *state-intention associations* as described above) based on useful abstract categories will come to be reinforced often and thereby strengthened. In this way, the system will learn to recognize abstract features of the state that are particularly helpful in determining which actions lead to intrinsic reward. Furthermore, because the representations of action intentions are themselves clustered and sequenced, the system may also acquire abstraction action plans that are also reinforced according to their intrinsic reward.

- *Exploitation of predictions admitted by thalamocortical sequence learning.* The previous section also described the sequence learning that continually operates over the learned categories; this sequence learning provides the functionality of *prediction* at multiple levels of abstraction (both semantic and temporal). This predictive capacity may then be exploited by the RL/reward system to significantly speed up learning in at least two different ways. First, the predictions as part of the internal state make the environment more observable to the agent. This can be a significant benefit because hidden state or equivalently partial observability significantly slows RL down (Singh et al. 2004). Second, the predictive capacity embodies an evolving probabilistic model of how the world evolves and can be used by the RL system to do "offline" learning or planning of state-action values which can lead to far better performance with far less "online" experience (Sutton 1990).

- *Parallel/cooperative reinforcement learning.* Each of TOSCA's segregated, multiple control loops is independently modulated by the intrinsic reward system. This structure makes TOSCA an instance of the more general class of *parallel reinforcement learning* systems, in which multiple, collaborating control systems interact through some shared state to maximize a shared reward signal. The key computational feature of such systems is that the explosive combinatorics of all the possible action combinations remains *implicit*: the control loops remain segregated, but cooperative behavior nevertheless emerges because each control loop adapts in the context of the behavioral consequences of the other co-adapting loops.

5.2.2 The role of intrinsic reward in shaping control

TOSCA's intrinsic reward system maps the rich internal state available to the agent to rewards that capture task-independent motivators (cf. Section 4.7) such as novelty, surprise, exploration, mastery (over environment), and like-me (by other agents in environment). These internal rewards lead the RL system in TOSCA to engage in exploration, play, and other behavior in the absence of explicit external reward (provided by some human specified task for example). As the agent engages in this behavior, the RL system learns a policy or rules of behaving that are captured in the form of the RL notion of options (Sutton et al. 1999). Options are temporally abstract actions that achieve subgoals and capture the intuitive notion of skills. Note that these skills could involve both external physical actions, for example an option could be about manipulating and mastering a physical object, as well as internal mental actions, for example an option could be about maintaining a particular episode of past experience in the internal state of the agent. These options or skills once learned become available as primitive actions to the agent and can then be chosen by the action generation and selection mechanism in the same way as pre-wired actions can be chosen by the RL system. Thus, more complex skills can be learned that use skills learned earlier as components. This allows the agent to incrementally learn a hierarchical set of skills that lead to increasing competence over its environment (Singh et al. 2005) and this

in turn makes the agent far more efficient at learning to solve externally specified tasks than would be possible without the internal reward based RL.

5.2.3 Control of motor system with multiple degrees of freedom

As discussed above there are multiple control loops in TOSCA and these loops allow both fine-grained control of individual degrees of freedom as well as coordinated control over multiple degrees of freedom in the form of motor routines. Some basic motor routines or skills will be pre-wired into the agent but many will be learned using the intrinsic reward based RL system outlined above. For example, driven by the internal motivation to achieve mastery over an object in its environment the parallel RL system described above would learn a complex motor routine or skill that orchestrates multiple control loops over time to reach for and manipulate that physical object as well as maintain internal state needed to accomplish the manipulation. Once learned, these skills that coordinate multiple control loops become available as primitive action choices to the parallel RL system leading to even more complex and richer hierarchical control of the motor system.

5.2.4 Control of cognitive resources

The rich multi-modal representational and memory systems described in Section 5.1 provide more than the basis for overt behavioral contingencies: they are themselves cognitive resources under adaptive control. The nature of this cognitive control ranges from the modulation of representations of current perceptions to the use of multi-modal imagery to simulate novel dynamic situations. In short, the system has control over aspects of its own internal state. This is a critical computational feature because it allows the system to move beyond reactivity to the kind of open-ended behavior that depends on arbitrary aspects of the past as well as the ability to flexibly project into the future. Figure 11 above depicts five critical classes of cognitive control loops which we briefly describe below. In all cases, what mediates top-down control is associations from frontal cortical areas (both to posterior regions and other frontal regions) that represent the selected cognitive actions. These frontal action representations are in turn contingent upon internal state and the frontal-striatal selection mechanisms described earlier.

Control of attention. The term "attention" has many meanings in psychology and cognitive neuroscience; we use it here to refer specifically to mechanisms of *selective activation and enhancement of perceptual representations*. For example, if it is adaptive for the system to attend to color features in certain situations, this may be accomplished by the selective enhancement of color features in lower-level perceptual representations which then bias the resulting higher-level categories to be more sensitive to color distinctions. Such changes in categorization then affect what new actions (internal and external) are selected next. In this way even relatively low-level attentional modulation has qualitative effects on behavior; the reward system reinforces those attentional contingencies that have positive effects.

Control of multi-modal working memory. Most tasks require the integration of information (either perceptual inputs or intermediate computational products) over time. There is considerable neural evidence that the persistence of such information over relatively short time periods (seconds to tens of seconds) depends critically on cortical representations (independent from the hippocampal subsystem) In TOSCA this persistent information is represented in the same higher-level perceptual posterior areas where the stimuli were originally processed. The information is maintained by excitatory connections from persistent (attractor-based) representations in prefrontal cortex; these prefrontal regions represent the "action" of maintaining a specific type of perceptual input. Those maintenance action contingencies that lead to intrinsic reward are reinforced. In this way TOSCA will learn to maintain task-relevant information in the face of potentially interfering irrelevant stimuli.

Control of multi-modal imagery and "simulators". Another emergent feature of TOSCA's architecture is the ability to reactive modality-specific states in the absence of sensory-motor stimuli. Such reactivations enable the conscious production of mental imagery in working memory (e.g., Kosslyn, 1980, 1994). Such reactivations further enable the less conscious (and unconscious) simulations that underlie a variety of high-level cognitive tasks. Substantial evidence now indicates that simulation supports conceptual processing, linguistic comprehension, memory retrieval, and thought. For reviews of supporting evidence in cognitive psychology, social psychology, cognitive neuroscience, and developmental psychology, see Barsalou (2003b), Barsalou, Simmons, Barbey, and Wilson (2003), Barsalou, Niedenthal, Barbey, and Rupport (2003), Niedenthal, Barsalou, Winkielman, Krauth-Gruber, & Ric (2005), Martin (2001), Pulvermüller (1999), Thompson-Schill (2003), Smith and Gasser (2005), and Thelen (2000).

Running SOCs in a top-down manner provides a natural mechanism for producing imagery and simulation. During learning, input from a modality-specific system produces a SOC that comes to represent this kind of input (Sections 4.2 and 4.3). Later, if the SOC is activated, it can in turn reactivate memories of the modality-specific states that produced it. These reactivations attempt simulate the states that the brain was in when it encountered instances of the SOC previously. Such simulations produce expectations that can serve a variety of computational functions, such as the completion of perceptions, the production of inferences during categorization that go beyond the information given, the representation of word meaning during language comprehension, the representation of memories during recollection, and the representation of ideas during thought.

Control of declarative/episodic memory retrieval. Although episodic encoding may be automatic, there is substantial psychological and cognitive neuroscience evidence that episodic memory retrievals are often under deliberate control. In TOSCA this controlled retrieval is realized via a combination of several mechanisms. Retrieval starts with the assembly of *retrieval cues* accomplished by the control of multi-modal working memory and imagery (see above; mediated primarily by frontal-posterior projections). These representations serve as cues by activating conjunctive episodic representations in the hippocampus via connections from posterior cortex to the hippocampus (see Section 4.5). But the hippocampus must be biased to process its inputs as cues for retrieval rather than new episodes to encode; this biasing happens via projections to the hippocampus from frontal regions that represent the specific intended cognitive action of retrieval [for precedents in the literature for such controlled retrieval see O'Reilly, Eichenbaum]. Again, those contingencies for retrieval actions that lead to intrinsic reward are reinforced, so that the system learns how to make effective use of its own episodic memory system.

Control of working memory for goals (abstract intentions). In the moment-to-moment control of behavior, the system faces the immediate problem of selecting the next best action to perform (across its multiple control loops). But as described above, this action selection may be contingent upon any feature of the internal state, including abstract features. Given that internal state is also under control—and crucially state that may *persist* for seconds to tens of seconds (see above)—the system can manipulate persistent internal state in order to direct its own behavior. In short, TOSCA will have the capacity to establish specific *goal representations* in working memory, and to learn when it is advantageous to do so. Functionally, such representations differ from the multi-modal working memory representations described above because they are exclusively about the *control* of future behavior and not about the temporary maintenance of relevant perceptual or imaginal information. Furthermore, the clustering over action sequences provides a suitable representational vocabulary for hierarchical goals because they are abstract intentions that already have associations with their specific constituent sequenced actions.

5.2.5 Social learning

In humans, learning rarely occurs save in the presence of other humans; from early development through adult stages, humans act predominantly in settings where other humans interact, largely via language. Interaction conditions include both direct learning from being told, as well as mixes of partially-guided and partially-independent discovery. In each case, balances must be struck between learning on one's own (via internal rewards and reinforcement learning) and learning from the social environment. Following are some of the ways social interaction will help speed up the reinforcement learning of skills and tasks. 1) The visual attention of the agent could be directed by the human guide towards socially salient or task/goal relevant stimuli, 2) The human guide should be able to demonstrate or suggest actions (using language) that coupled with the intrinsic reward for being liked would lead to the agent to explore those actions, 3) The human guide may provide explicit reward to the reinforcement learning system inside TOSCA, and 4) The human guide may provide subgoals by pointing out landmark states or other novel states relevant to the overall task faced by the agent. All of these forms of social interaction directly impact the reinforcement learning system by either providing internal or external reward or suggesting salient states to pay attention to or providing salient actions to imitate. Together, these can significantly reduce the amount of data needed by the agent to learn competence over its environment.

5.2.6 Language and control

The previous section highlighted the important role of language in social learning. Here we consider one specific avenue through which language has a dramatic impact on cognition and behavior: instruction taking. The ability to take instructions is computationally significant because it means that the system is *immediately taskable*, providing a way to move past the incremental adaptation that is characteristic of reinforcement learning. In this section we provide a sketch of how the architectural mechanisms in TOSCA will support instruction taking, focusing on the situation where the system acts immediately upon its received linguistic input. (Instruction taking situations involving greater temporal separation of the instruction and behavior would tap into the same mechanisms described here, but would additionally involve the long term declarative memory system).

At a high level, instruction taking can be decomposed functionally into (a) *comprehension* of the linguistic input, which yields representations that are (b) *interpreted* to produce behavior. (These separate functions need not be strictly staged but may be tightly interleaved in the process of incremental understanding and behaving). The representational and functional capacities of TOSCA described above are sufficient to accomplish these functions and, crucially, to accomplish these functions with novel linguistic inputs. Briefly, the process would work as follows:

- *Comprehension* happens as the linguistic input is incrementally processed and given a hierarchical representation as described above in Section 5.1 (which outlined how certain aspects of linguistic grammars may emerge from clustering and sequencing). The critical feature is that the linguistic input is represented at multiple levels of abstraction in a way governed by the (learned) grammar of the language, making explicit the structural (and thus indirectly the semantic) relations among the constituents parts of the input.
- Task-critical aspects of these representations are *selectively maintained* as an assembly of cortical representations in the frontal inferior and posterior language areas so that they may guide future behavior (see Section 5.2.4 above on adaptive control of working memory). This provides an important link between comprehension and interpretation.
- The *interpretation* of the instructions happens as these linguistic representations function as goals upon which action selection (in the multiple parallel loops) is contingent. Such linguistically-driven action selection is learned via the same reinforcement learning mechanisms described earlier. A crucial aspect of this learning is that it exploits the abstract generalizations admitted by the hierarchical representations (see section 5.2.1). This generalization, coupled with the fact that the interpretation takes place incrementally, will give rise to an interpretive skill that will transfer immediately to novel linguistic inputs that share critical structure. For example, the interpretation of simple instructions taking the form *perform a specific action upon an object* will consist of separate acts of orienting and grasping etc. that will be useful for a wide range of different specific actions. The most-often rewarded contingencies for such initial orienting will thus tend to abstract away (via the clustered representations) from the specific instructed actions. Similarly, contingencies for aspects of the control to accomplish the instructed action may abstract away from the specific linguistic label used to identify the object, because such abstract contingencies are reinforced more often.

Thus TOSCA's ability to take instructions will arise from an interaction of the emergent, generative grammatical representations described above and the reinforcement-learning-based interpretation of linguistic representations that exploits the learned abstract grammatical categories. The reinforcement learning of the interpretive skill is part of the gradual, incremental process of language acquisition, but it crucially yields immediate taskability. In short, language comprehension and use is a procedural skill operating on special types of representations, but acquired via general processing principles. This novel approach to instruction taking has the virtue that learning permeates every aspect of the processing, and we believe it will provide a compelling demonstration of the cognitive power that emerges from the interaction of TOSCA's basic architectural mechanisms.

6. Conclusion

The goal of this paper is to describe our design for a new cognitive architecture based on the brain: TOSCA. With many cognitive architectures, a design is a straight forward description of component modules (such as procedural memory, working memory, bottom-up sensor processing), and a straightforward mapping of components onto high-level functionality. But in designing the brain, evolution took an interesting turn. Instead of having a parallel decomposition of functionality and structure, it built on a more primitive set of computational components that we presented in Figure 1. These primitives are tightly interconnected and form circuits (Section 4), so that functionality emerges from the interactions of these multiple circuits (Section 5).

This organization starts to answer the question as to where is the “magic in human cognition” – it is not in any one module that AI hasn’t yet discovered, but it is in choosing the right set of building blocks and the connections between them. But above and beyond the static organization is the dynamics of the system where learning is ubiquitous. For learning to be successful, the right information must be made available, and this organization brings the information to the right places. For some components the learning is mostly bottom-up, as the system learns statistical regularities of its environment (as in learning clusters and sequences). For others it is associational where learning brings together co-occurring sensations across modalities. And for action, intrinsic reward drives the learning of control across not just external actions, but the system’s control of itself. This is the path for TOSCA, and the path to a new generation of brain-based cognitive systems.

References for PART I

Alexander GE, DeLong MR, Strick PL. (1986). Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Annual Review of Neuroscience* **9**, 357–38.
Alexander GE, DeLong MR, Strick PL. (1986). Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Annual Review of Neuroscience* **9**, 357–38.
Arbuthnott GW, MacLeod N, Rutherford A. The rat cortico-striatal pathway in vitro. *Journal of Physiology London* **367** (1985), 102P.
Arbuthnott GW, MacLeod N, Rutherford A. The rat cortico-striatal pathway in vitro. *Journal of Physiology London* **367** (1985), 102P.
Barsalou, L.W. (1987). The instability of graded structure: Implications for the nature of concepts. In U. Neisser (Ed.), *Concepts and conceptual development: Ecological and intellectual factors in categorization* (pp. 101-140). Cambridge: Cambridge University Press.
Barsalou, L.W. (1989). Intraconcept similarity and its implications for interconcept similarity. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 76-121). Cambridge: Cambridge University Press.
Barsalou, L.W. (1993). Flexibility, structure, and linguistic vagary in concepts: Manifestations of a compositional system of perceptual symbols. In A.C. Collins, S.E. Gathercole, & M.A. Conway (Eds.), *Theories of memory* (pp. 29-101). London: Lawrence Erlbaum Associates.
Barsalou, L.W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22*, 577-660.
Barsalou, L.W. (2003a). Abstraction in perceptual symbol systems. *Philosophical Transactions of the Royal Society of London: Biological Sciences, 358,* 1177-1187.
Barsalou, L.W. (2003b). Situated simulation in the human conceptual system. *Language and Cognitive Processes,* 18, 513-562.
Barsalou, L.W. (2005). Abstraction as dynamic interpretation in perceptual symbol systems. In L. Gershkoff-Stowe & D. Rakison (Eds.), *Building object categories* (389-431). Carnegie Symposium Series. Mahwah, NJ: Erlbaum.
Barsalou, L.W., Niedenthal, P.M., Barbey, A., & Ruppert, J. (2003). Social embodiment. In B. Ross (Ed.), *The Psychology of Learning and Motivation,* Vol. 43 (pp. 43-92). San Diego: Academic Press.
Barsalou, L.W., Simmons, W.K., Barbey, A.K., & Wilson, C.D. (2003). *Grounding conceptual knowledge in modality-specific systems. Trends in Cognitive Sciences, 7,* 84-91.
Cree, G. S,. & McRae, K. (2003). Analyzing the factors underlying the structure and computation

of the meaning of chipmunk, cherry, chisel, cheese, and cello (and many other such concrete nouns). *Journal of Experimental Psychology: General, 132,* 163-201.

Damasio, A.R. (1989). Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. *Cognition, 33,* 25-62.

Farah, M. J., & McClelland, J. L. (1991). A computational model of semantic memory impairment: Modality specificity and emergent category specificity. *Journal of Experimental Psychology: General, 120*, 339-357.

Gluck, M. A., Myers, C. E., & Thompson, R. F. (1994). A computational model of the cerebellum and motor-reflex learning. In S. Zournetzer, J. Davis, T. McKenna, & C. Lau (Editors). An Introduction to Neural and Electronic Networks (Second Edition). 91-80.

Gluck, M.A., Allen, M.T., Myers, C.E., & Thompson, R.F. (2001) Cerebellar substrates for error-correction in motor-reflex conditioning. Neurobiology of Learning and Memory, 76, 314-341.

Kemp JM, Powell TPS. The corticostriate projection in the monkey. *Brain* **93** (1970), 525-546**.**

Kemp JM, Powell TPS. The corticostriate projection in the monkey. *Brain* **93** (1970), 525-546**.**

Kosslyn, S.M. (1980). *Image and mind.* Cambridge, MA: Harvard University Press.

Kosslyn, S.M. (1994). *Image and brain.* Cambridge, MA: MIT Press.

Martin, A. (2001). Functional neuroimaging of semantic memory. In R. Cabeza & A. Kingstone (Eds.), *Handbook of functional neuroimaging of cognition* (pp. 153-186). Cambridge, MA: MIT Press.

McRae, K., de Sa, V.R., & Seidenberg, M.S. (1997). On the nature and scope of featural representations of word meaning. *Journal of Experimental Psychology: General, 126,* 99-130.

Montague PR, Dayan P, Sejnowski TJ. A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of Neuroscience*, **16** (1996), 1936–1947.

Montague PR, Dayan P, Sejnowski TJ. A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of Neuroscience*, **16** (1996), 1936–1947.

Niedenthal, P.M., Barsalou, L.W., Winkielman, P., Krauth-Gruber, S., & Ric, F. (2005). Embodiment in attitudes, social perception, and emotion. *Personality and Social Psychology Review, 9,* 184-211.

Pulvermüller, F. (1999). Words in the brain's language. *Behavioral and Brain Sciences, 22,* 253-336.

Romo R, Schultz W. Dopamine neurons of the monkey midbrain: contingencies of responses to active touch during self-initiated arm movements. *The Journal of Neurophysiology* **63** (1990), 592-606.

Romo R, Schultz W. Dopamine neurons of the monkey midbrain: contingencies of responses to active touch during self-initiated arm movements. *The Journal of Neurophysiology* **63** (1990), 592-606.

Schultz W. Responses of midbrain dopamine neurons to behavioral trigger stimuli in the monkey. *The Journal of Neurophysiology* **56** (1986), 1439-1461.

Schultz W. Responses of midbrain dopamine neurons to behavioral trigger stimuli in the monkey. *The Journal of Neurophysiology* **56** (1986), 1439-1461.

Schyns, P.G., Goldstone, R.L., & Thibaut, J.P. (1998). The development of features in object concepts. *Behavioral and Brain Sciences, 21,* 1-54.

Simmons, W.K., & Barsalou, L.W. (2003). The similarity-in-topography principle: Reconciling theories of conceptual deficits. *Cognitive Neuropsychology, 20,* 451-486.

Smith, L., & Gasser, M. (2005). The development of embodied cognition: Six lessons from babies. *Artificial Life, 11*, 13-29.

Sutton, R.S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. Proceedings of the Seventh International Conference on Machine Learning, pp. 216-224, Morgan Kaufmann.

Sutton, R.S., Precup, D., Singh, S. (1999). Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. Artificial Intelligence 112:181-211

Thelen, E. (2000). Grounded in the world: Developmental origins of the embodied mind. *Infancy, 1*, 3-30.

Thompson-Schill, S.L. (2003). Neuroimaging studies of semantic memory: inferring "how" from "where". *Neuropsychologia, 41*, 280-292.

D. E. Berlyne. A theory of human curiosity. British Journal of Psychology, 45:180–191, 1954.

D. E. Berlyne. Conflict, Arousal and Curiosity. McGraw-Hill, N.Y., 1960.

D. E. Berlyne. Aesthetics and Psychobiology. Appleton-Century-Crofts, N.Y., 1971.

D. O. Hebb. The Organization of Behavior. Wiley, N.Y., 1949.

G. Di Chiara. Drug addiction as dopamine-dependent associative learning disorder. European Journal of Pharmacology, 375(1-3):13–30, 1999.

G. Barto. Adaptive critics and the basal ganglia. In J. C. Houk, J. L. Davis, and D. G. Beiser, editors, Models of Information Processing in the Basal Ganglia, pages 215–232. MIT Press, Cambridge, MA, 1995.

K. J. Friston, G. Tononi, G. N. Reeke, O. Sporns, and G. M. Edelman. Value-dependent selection in the brain: Simulation in a synthetic neural model. Neuroscience, 59:229–243, 1994.

S. Kakade and P. Dayan. Dopamine bonuses. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, Advances in Neural Information Processing Systems 13, pages 131–137. MIT Press, 2001.

S. Kakade and P. Dayan. Dopamine: Generalization and bonuses. Neural Networks, 15:549–559, 2002

Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Proceedings of the Sixteenth International Conference on Machine Learning. Morgan Kaufmann, 1999.

P. R. Montague, P. Dayan, and T. J. Sejnowski. A framework for mesencephalic dopamine systems based on predictive hebbian learning. Journal of Neuroscience, 16:1936–1947, 1996.

P. Dayan and B. W. Balleine. Reward, motivation and reinforcement learning. Neuron, 36:285–298, 2002.

P. Dayan and T. J. Sejnowski. Exploration bonuses and dual control. Machine Learning, 25:5–22, 1996.

P. Dayan and A. J. Yu. Uncertainty and learning. IETE Journal of Research, 49:171–182, 2003.

S. Kakade and P. Dayan. Dopamine bonuses. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, Advances in Neural Information Processing Systems 13, pages 131–137. MIT Press, 2001.

S. Kakade and P. Dayan. Dopamine: Generalization and bonuses. Neural Networks, 15:549–559, 2002.

M. S. Hooks and P. W. Kalivas. Involvement of dopamine and excitatory amino acid transmission in novelty- induced motor activity. Journal of Pharmacology & Experimental Therapeutics, 269:976–988, 1994.

J. C. Horvitz, T. Stewart, and B. Jacobs. Burst activity of ventral tegmental dopamine neurons is elicited by sensory stimuli in the awake cat. Brain Research, 759:251–258, 1997.

W. Schultz. Predictive reward signal of dopamine neurons. Journal of Neurophysiology, 80:1–27, 1998

W. Schultz, P. Dayan, and P. R. Montague. A neural substrate of prediction and reward. Science, 275:1593–1598, March 1997.

R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112:181–211, 1999.

J. Yu and P. Dayan. Acetylcholine in cortical inference. Neural Networks, 15:719–730, 2002.

S. Singh, M. R. James, and M. R. Rudary. Predictive State Representations: A New Theory for Modeling Dynamical Systems In Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI), pages 512-519, 2004.

S. Singh, A. G. Barto, and N. Chentanez. Intrinsically Motivated Reinforcement Learning. In Proceedings of Advances in Neural Information Processing Systems 17, 2005.

PART II: STORM Framework

The STORM Framework is a toolkit that will be used to build one or more specific TOSCA architectures. The Framework should be general purpose and flexible so it supports the design and experimentation on particular TOSCA architectures. This means the focus for the Framework is on generality and flexibility of infrastructure, while a particular architecture might be much more tightly constrained and make specific behavioral predictions.

To ensure that any TOSCA architecture makes strong biologically-inspired commitments, the STORM Framework imposes a set of constraints on the architectural design. First, all processing must be mapped to a specific brain region. This ensures that an executing TOSCA system always makes a formal commitment to how all simulated processing would map to biological processing. Second, all communication between modules, within the simulation, must conform to known biological constraints based on the current literature. If the connections made within a TOSCA simulation violate the known connectivity properties of regions within the brain, these constraint violations will be explicitly detected by the Framework.

In order to support the expression of these biological constraints, the processing within the Framework must have clearly defined boundaries. The proposal is to divide the elements of the Framework into two categories:

- Function Modules
- State Variables

Function modules perform processing and state variables are used for communication between modules. Modules receive inputs from a set of state variables and generate outputs to one or more state variables.

The critical constraints are that values can only be persistently stored within state variables and all state variables must be mapped to a brain region. Together, these constraints imply that for any model implemented within the STORM Framework we can determine how the model is connected and what commitments the model makes for regions of the brain where processing occurs and how those brain regions are connected. Those model commitments can then be reviewed against current knowledge of biological processing to determine the quality of the fit.
The relationships between function modules and state variables and how they map to biological processing are represented through three logical graph structures within the STORM Framework:

1. Functional Connectivity Graph
2. Brain Mapping Graph
3. Brain Connectivity Graph

Functional Connectivity Graph

This graph establishes a mapping of state variables to input and outputs of function modules. It defines the flow of data through the simulation.

For example, Figure 14 shows module M1 receiving inputs from state variables A, B and C and generating outputs for A and D. Module M2 receives input from D and generates output for C.

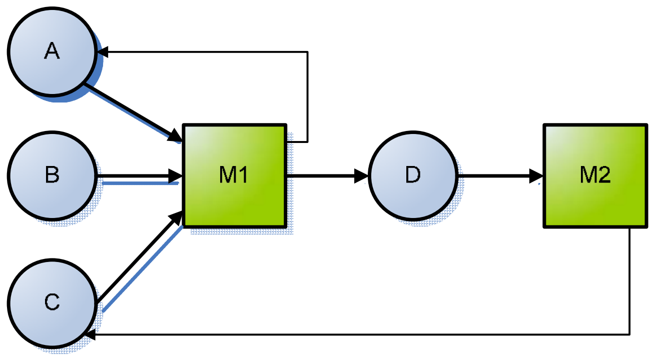


Figure 14. Functional Connectivity Graph

This graph is implicitly defined by the inputs and outputs from function modules. It is not explicitly represented in the Framework as a separate data structure, although we will have tools that can extract this graph for display or analysis based on examining the connections between modules and state variables.

State variables can also be used to provide private storage where only a single function module ever accesses the state variable. Function modules are potentially executing asynchronously and at a range of time scales.

Some key properties of this graph are:

- **Flexible replacement of modules and state variables**
  A module or state variable can be replaced with a different implementation with minimal impact on other modules and variables. This supports experimentation by swapping in different implementations for parts of the overall architecture. It also supports incremental and distributed development of the entire architecture as placeholders can be used until a full component has been developed and can replace the placeholder.

- **State variables are used for all persistent data**
  All data that persists from one cycle of the simulation to another (i.e. that is not temporary storage used in a calculation) should be represented by a state variable.
  Some examples of state variables are:
    - A frame buffer of video input
    - Connection strengths in a neural network
    - Variables used to control search
    - Matrix or vector of floating point values
    - Production rules (condition -> action rules)

  This property relates to the brain mapping graph (below). Only state variables would be mapped onto brain regions. There would be no need to map function modules to brain regions explicitly as that mapping would be implicit in the state variables.

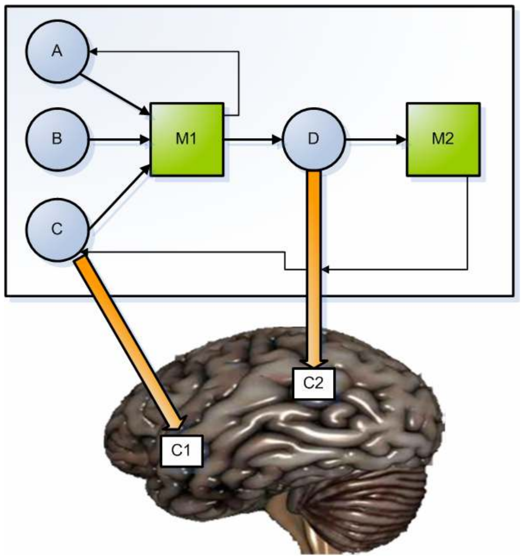
Brain Mapping Graph

The brain mapping graph establishes how function modules and state variables are mapped to different regions of the brain. In the current design only state variables are explicitly mapped to brain regions. Function modules are mapped to regions based on the state variables they use (Figure 15).

Unlike the functional connectivity graph, the brain mapping graph will be explicit. There will be a specific data structure in the Framework that defines how each state variable is mapped to a region of the brain.
The mapping:

- **Will be complete**
  All state variables must map to some brain region. This ensures that there's an explicit mapping from a running simulation to the brain.

- **May include unspecified regions**
  A state variable may have an undetermined location but this must be explicitly indicated in the mapping.



F

Figure 15. Brain Mapping Graph

Brain Connectivity Graph

The functional connectivity graph together with the brain mapping graph imply a brain connectivity graph (e.g. Figure 16).

That is, by examining how the state variables and function modules are connected together and how they are mapped to brain regions makes a prediction about how the brain regions are connected. This graph can then be tested against known constraints for how brain regions are actually connected. These constraints will be explicitly represented within the framework and the brain connectivity graph will be deduced automatically from the brain mapping graph and the functional connectivity graph.

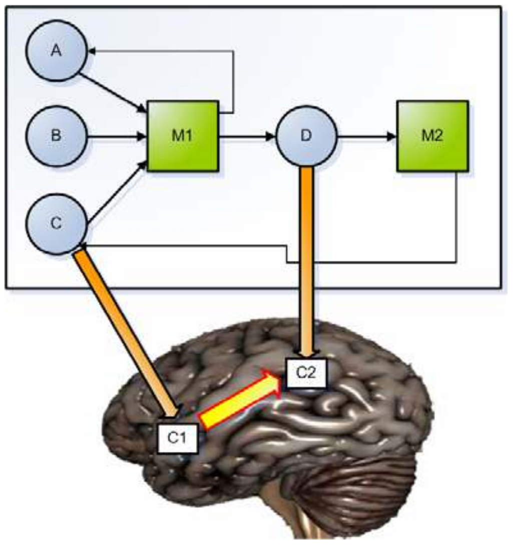


Figure 16. Brain Connectivity Graph

Function Modules and Time

Function modules may execute asynchronously and at different time scales. Two communicating modules may be executing on a single machine or across a network between a cluster of machines. The STORM Framework provides an abstraction layer over these details, allowing each module (and set of state variables) to be defined in terms of its own temporal constraints, with the Framework handling the details of communication and synchronization between modules and variables.

Figure 17 shows an example of the capabilities provided by the Framework. Modules M1 and M2 are executing on a 1ms and 5ms time scale respectively. Every 50ms M2 posts a result to the state variable D which is consumed by M3. This synchronization is achieved through two different clocks – T1 is the master clock for the simulation and T2 is a local clock used only to synchronize the behavior of M1 and M2.

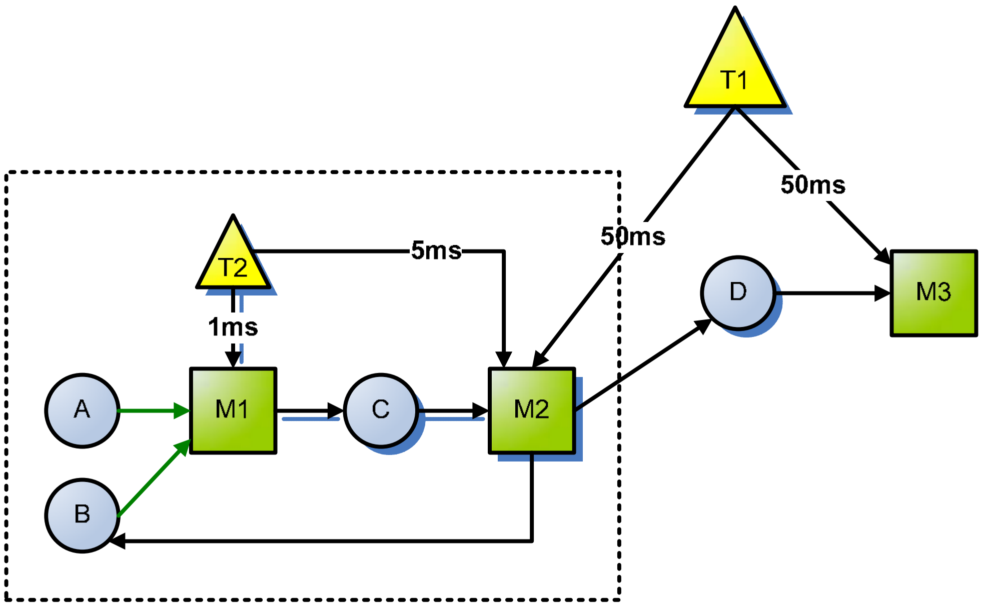


Figure 17. Local Clock Example

Each module expresses the constraints it is placing on the overall flow of time through the simulation by making calls an appropriate clock. Based on these constraints, the clock(s) determine when simulation time can advance and by how much before further processing must occur in a module. Each module is unaware of the complexity within the entire system and deals only in the local constraints that concern it (e.g. M2 reads input from C every 5ms and outputs to D every 50ms). Additionally, everything within the dotted line shown in Figure 17 could be collected into a larger function module. This allows hierarchies of modules to be built and used in experimentation. For instance, two different implementations of a particular function module (the first a simple module and the second a complex collection of submodules, variables and clocks) could be tested against each other without modifying the rest of the simulation. The sample code below shows the detail of how a function module would appear within the Framework if written in C++ (the actual Framework will support multiple implementation languages):

```
 while (!m_Quit)
{

// Wait for a signal from the clock (if time driven)

// or from an input changing (if event driven)

WaitForNextSignal() ;


// Process any events we were sent while sleeping

ProcessEvents() ;


// Get the current clock time. This value can't change until

// we signal that it's ok for the clock to advance.

Time time = GetClock()->GetTime() ;


// Get the value of the inputs at the current time

Value a = GetInput("A")->GetValue(time) ;

Value b = GetInput("B")->GetValue(time) ;

Value c = GetInput("C")->GetValue(time) ;


// We've read our inputs so the clock can continue

// but it can't go beyond the time when we generate output.

// Changing the boundary time allows the clock to advance.

GetClock()->ClockCannotAdvanceBeyond(this, time+4) ;


// Perform a calculation based on the inputs (could be long time)

Calc(&a, &b, &c, &result) ;
// Generate output 5ms after this function module was triggered

GetOutput("A")->SetValue(&result, time+5) ;


// Set up our next triggering event

GetClock()->RegisterWakeup(this, time+10) ;


// Clock can now advance freely up to our next trigger event

GetClock()->ClockCannotAdvanceBeyond(this, time+10) ;

}
```

The module initially waits for a signal (either a specific amount of time passing or an input variable changing) before beginning processing. It receives any notifications of new system events before reading the current input values for state variables A, B and C. While reading its inputs the simulation clock is kept at a known time, ensuring that the inputs remain valid. Once the module has read all of its inputs it signals that the clock can now advance but not beyond the point when the module will generate output. The module computes the output and posts the new value (in this case in a feedback loop to variable A, 5ms after it awoke). It then requests its next wakeup signal—in this example by asking the clock to signal it when 10ms have passed and releases the clock to move up to that new time.

This example shows a module that is *time-driven*, where it's processing is triggered by the flow of time. A very similar logic is used when the module is *event-driven*, where it's processing is triggered by the change of an input. The module registers in advance with the input variable to request notification when the variable changes state and then engages in similar processing to the time-driven example shown here.

The Framework maintains the actual location of all state variables, which could be on different physical machines, as well as determining the correct flow of simulated time within the constraints each module imposes. The specific location of a state variable can be changed without affecting existing modules and the details of how a module is implemented can be changed without affecting other modules as long as its inputs and outputs are not significantly changed.

Implementation Flexibility

The Framework will allow modules to be written in a variety of different languages and then combined together to form a complete model. This is achieved by implementing the core functionality in C/C++ and then machine generating interfaces in other languages. We have other projects that have adopted this method and support 5+ languages in this manner. This cross-language capability allows each researcher or team to adopt the language that best supports their work and yet still create an integrated system.

The Framework will also support a range of runtime configurations, from a single process executing the entire simulation, to a series of processes using shared memory within a single machine and up to a cluster of machines each executing a part of the entire simulation. This flexibility is achieved by basing all of the communication on message passing between state variables and then abstracting over the details of how those messages are passed—e.g. by accessing a local pointer or by sending a message over a socket to a different machine. This abstraction allows the hardware infrastructure to scale up as the requirements of a particular task increase, without having to redesign modules or have different implementations for different runtime configurations.

Example Task Requiring Learning and Memory

In order to demonstrate the Storm framework, we created a simple task that requires learning control knowledge for both internal and external actions. The example task is set in a 5x5 grid-world, shown in Figure 18. The domain contains three special locations, or *boxes,* in fixed positions: boxes A and B are *reward* boxes, while box M is an *information* box. The agent is rewarded with a positive reward when it opens one of the boxes and a negative reward when opening the other. The agent perceives a symbol when it opens the information box; this symbol is correlated with the location of the positive reward box (but does not correlate to any perceived feature of the boxes). An agent that cannot maintain the symbol in an internal memory would be unable to receive the maximum reward in every episode, making the task un-learnable.

The agent can move in the four cardinal directions, and if a box is in its current location, the agent can open the box. The agent perceives its location in the grid and any reward signal, but cannot perceive the labels on the boxes (A or B). If the agent is in the information box square and the box is open, the agent also perceives a symbol. An episode concludes when the agent opens the box containing the positive reward. The location of the rewards is randomized between episodes.

Reward is structured such that a positive reward has magnitude of +10, a negative reward is -10, and on every step that the agent does not open a reward box, it receives -1 reward.

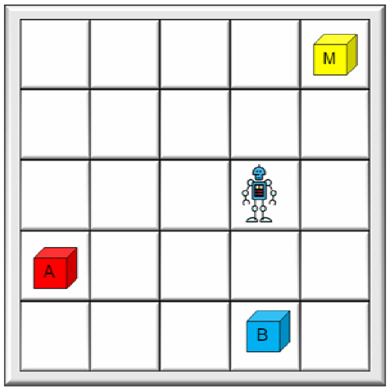


Figure 18. Information Box Task

An Example Architecture Developed using Storm

In order to help illuminate some of the framework's capabilities, we used Storm to develop a simple architecture capable of supporting an agent that learns to perform in the example task. This architecture combines a simple long-term memory with a basic reinforcement learning mechanism that learns control knowledge for both internal and external actions.

Our example architecture is shown in Figure 19. Function modules in the figure are represented as rectangles, and state variables as ovals. In this model, the environment is represented as a function module (for convenience) which receives a motor action as input and generates sensory information as output. Sensory Input is used by both Long and Short Term memories, which in turn is used by Action Selection to choose an internal Long Term Memory retrieval as well as an external Motor action. The Reinforcement Learning mechanism uses Working Memory, the Internal Reward Signal, and selected actions to adjust the control knowledge used by Action Selection.

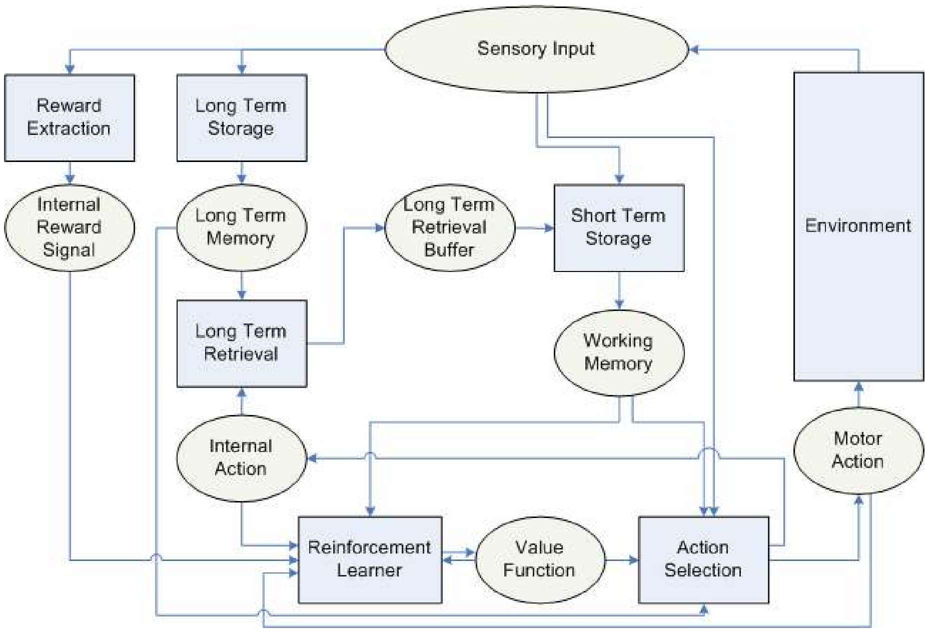


Figure 19. Simple Architecture Implemented in Storm

The details of each function module are described below. The Reward Extraction module reads the explicit reward generated by the environment in Sensory Input and stores it as an Internal Reward Signal. Long Term Storage stores any perceived symbol (i.e. the contents of the information box) to Long Term Memory. Long Term Retrieval retrieves a symbol corresponding to the Internal Action from memory and stores it in the buffer. Short Term Storage reads the agent's location from Sensory Input and the contents of the Long Term Retrieval Buffer and puts the concatenation of both in Working Memory.

The agent decides how to act in the Action Selection module, which uses Working Memory and the Value Function to select its actions (using a decaying epsilon-greedy strategy). The Value Function is a table that associates a pair of internal and motor actions with the contents of working memory and the estimated future reward of applying those actions. The Value Function is adjusted by the Reinforcement Learner with Sarsa (Sutton, 1996) based on input from Working Memory, Internal Reward Signal, and Internal and Motor Actions.

In the example architecture, function modules initiate their processing when their input state variables change, and all take a fixed amount of time to process and create results. During execution, many of the modules will execute in parallel, such as those that depend on Sensory Input. Others execute in sequence because of the dependencies of their input variables on other function modules. This parallelism enables an agent to perform internal and motor actions simultaneously.

An execution trace of the example architecture's function modules is shown in Figure 20, which is generated from the execution logs by a Storm utility. There are four different types of events logged by Storm for a function module.

- *RequestWakeup*: can occur on the time step when an input variable's value changes,
- *Wakeup*: occurs on the time step immediately following a *RequestWakeup* event (unless explicitly delayed),
- *Finished*: occurs on the time step on which the module sets its output variables and completes processing, and
- *Processing*: this is the time that a module is inferred to be processing between *Wakeup* and *Finished* events.

Multiple events that occur on the same time step are plotted as one event (e.g. all Environment events occur on the same step).

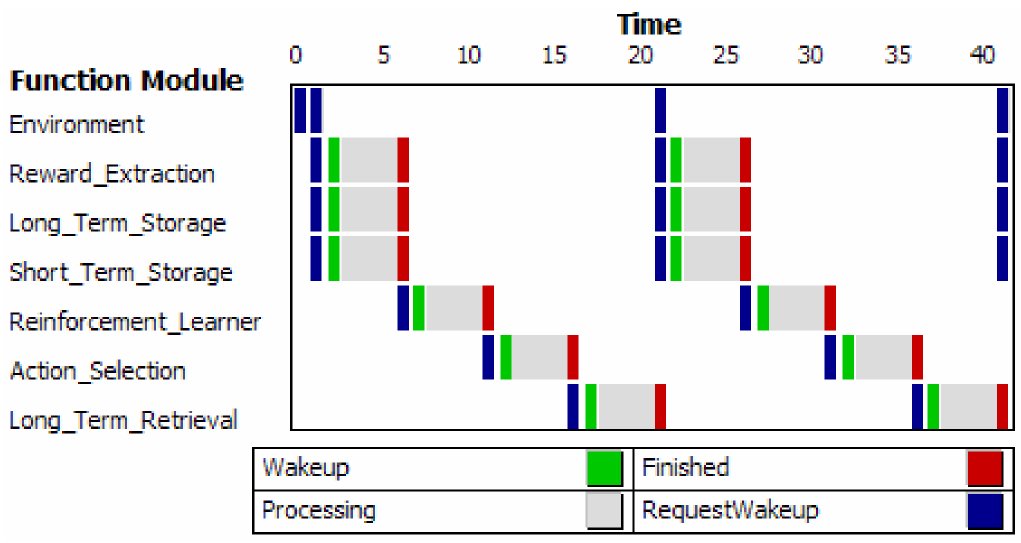


Figure 20. Sample execution trace of the example architecture as generated by a Storm utility

Processing begins in the Environment module which sets the Sensory Input state variable (see Figure 19). When Sensory Input is set, a *RequestWakeup* event triggers the Reward Extraction, Long Term Storage, and Short Term Storage modules. All three modules then process in parallel, after which they set their respective output variables.

The Reinforcement Learner module next begins processing, as it relies on the Internal Reward Signal set by the Reward Extraction module. Similarly, the Action Selection module relies on the output of the Reinforcement Learner, and the Long Term Retrieval module on Action Selection, which explains the serial behavior seen in Figure 20. This behavior arises from the dependences of the input variables of each module, and is not explicitly timed or engineered. However, the Environment module *is* configured to process periodically, which explains why it does not begin executing at the same time as the Long Term Retrieval module even though inputs for both modules are set by Action Selection.

Although the mapping of state variables to brain regions is an important commitment made in Storm, this example architecture is so simple that we do not hypothesize a mapping. Rather, the purpose of this example is to illustrate the framework's specification and simulation capabilities.

Results

Example Architecture

We developed two agents in the architecture to perform the example task, one that automatically retrieves the information symbol from long-term memory (when available) and one that must learn to retrieve it. The performance of the two agents is shown in Figure 21. Asymptotically, the behavior of both agents is the same: the agent moves directly to the information box, opens it, and then simultaneously retrieves the identifying symbol from long-term memory while navigating to the positive reward box, opening it upon reaching its location. The results indicate that learning both control knowledge for an internal action in addition to an external motor action is not significantly more difficult than for an external action alone.

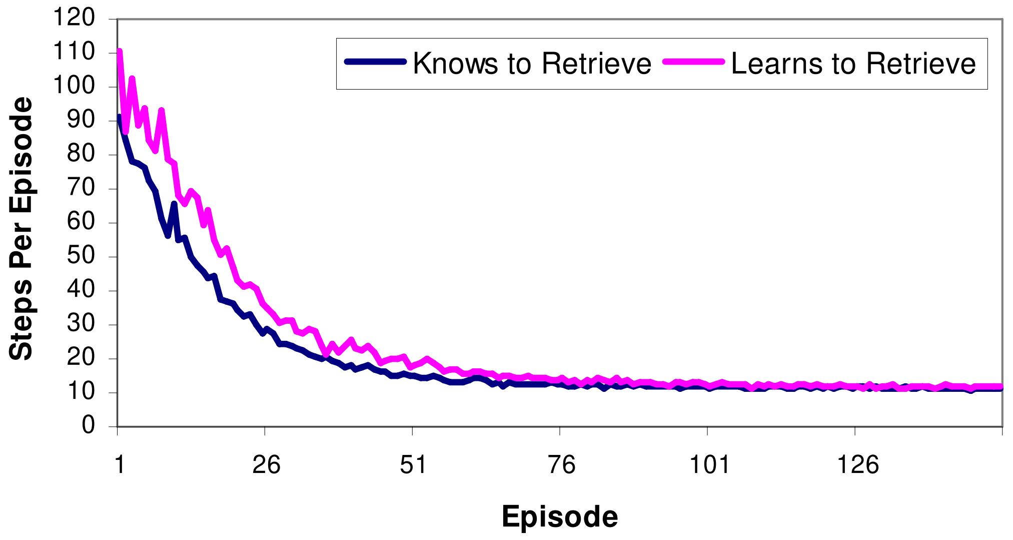


Figure 21. Learning curves for two agents performing on the simple task, average of 225 trials

Modified Task and Expanded Working Memory

To study the flexibility of an architecture using the Storm framework, we modified the task so that the agent had to learn to manage long-term memory retrievals:

- Instead of a single motor action to open a box, the agent now has two available actions. When the correct one is used to open the positive reward box, the standard reward is still received. However if the reward box is opened with the other action, a smaller positive reward (+1) is received.

- The information box contains an additional symbol identifying the correct action to use when opening the positive reward box. Both symbols are still automatically stored to long-term memory.

After an agent using the example architecture opens the information box, both symbols are then automatically stored to long-term memory. However, the Long Term Retrieval Buffer (and thus Working Memory) can still only store one retrieved symbol at a time. The agent therefore must learn to recall the two symbols at different times: the symbol identifying the correct box during navigation and the action symbol on the step before it will open the box.

We tested an agent using the example architecture as well as an agent with an expanded working memory that can store the two most recently retrieved symbols in working memory on the modified task. The results for both agents are presented in Figure 22. Although the agent using the architecture modified with an expanded working memory learns more quickly than the agent using the unmodified architecture, these results show that an agent using the unmodified architecture with limited working memory is still able to learn the modified task.

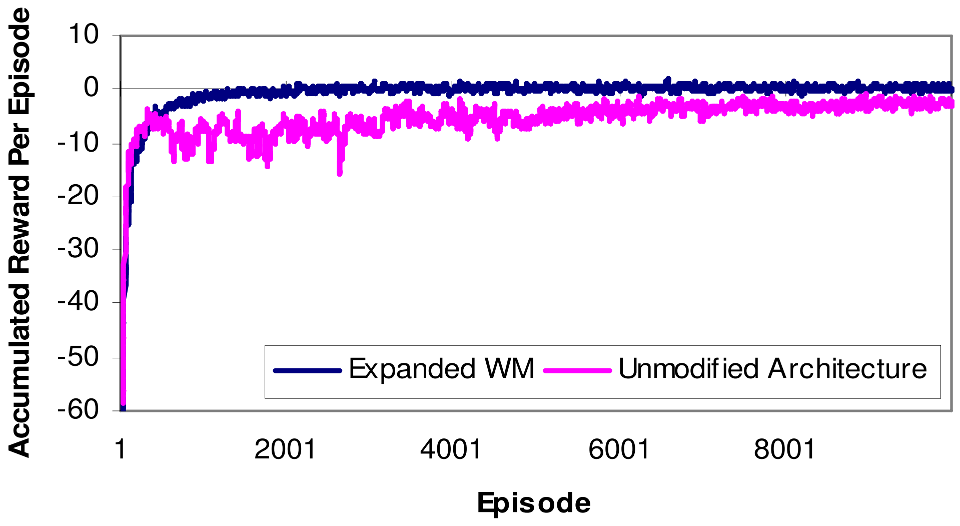


Figure 22. Learning curves for agents performing on the modified task, 25 per. moving avg. of medians for 45 trials

In order to modify the architecture with an expanded working memory, only the Short Term Storage function module and Working Memory state variable needed to be changed – the rest of the architecture's function modules and state variables remained the same. This demonstrates an advantage to experimenting within the Storm framework: the modularized approach to development leads to architectures that can be modified quickly and easily.

Architectural Delay

In our example architecture, all function modules took the same constant amount of time to process data (5 units of time as seen in Figure 20). In order to experiment with function modules processing at different time scales, we introduced a delay to the Long Term Retrieval module: with a delay, the module processes for 20 units of time rather than 5. This change has two effects: first, retrieved memories are available two environment steps after the Internal Action is selected; second, retrieved symbols in the buffer persist for two environment steps. Because of these changes, the agent can improve its performance by learning to make a retrieval from Long Term Memory two steps before it gets to the reward box.

The results of two agents, one modified with a delayed retrieval and the other unmodified, in the modified task are shown in Figure 23. While the agent using the unmodified architecture initially learns more quickly, the behaviors are indistinguishable after the 2000th episode.

In order to experiment with delaying Long Term Retrieval in the architecture, our implementation in Storm required only a single line of code to be changed. Storm's mechanism for scheduling the processing of function modules makes changing timing constraints to be a straightforward exercise.

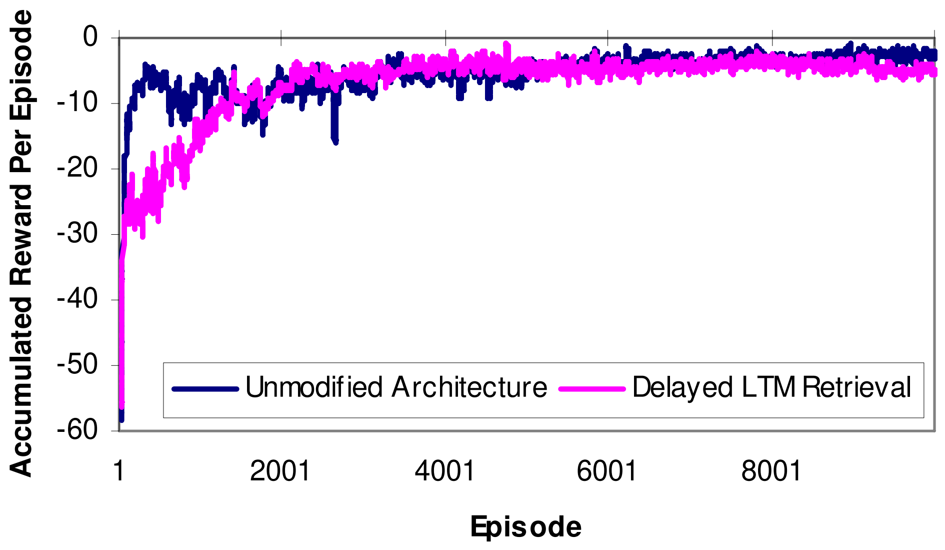


Figure 23. Learning curves for an architecture modified with delayed Long Term Retrieval compared with an unmodified architecture, 25 per. moving avg. of medians for 45 trials

Summary of Experiments

The Storm framework has allowed us to experiment with the example architecture in several dimensions, the results of which are not all shown in this paper:

1. We experimented with two timing conventions: both waking function modules when input variables have been set as well as function modules processing periodically at set intervals. The example architecture implements a hybrid approach and uses both approaches in its modules.

2. We experimented with the timing of individual modules, delaying their output such that the processing time of various modules overlaps.

3. We explored reinforcement learning modules implementing a variety of learning algorithms with various parameter settings; switching algorithms is as simple as changing the module used by the framework.

4. We have simulated environments in C++ function modules and interfaced to external Java environments.

When experimenting along all of these dimensions, the necessary changes to function modules were minor and no changes to the framework were necessary. In contrast, experimenting with existing cognitive architectures to modify the behavior of working memory, long-term memory, or timing constraints can often be difficult and time consuming.

Discussion

By developing our example architecture using the Storm framework, we have had valuable experiences which begin to shed light on the advantages (and disadvantages) of using a lightweight framework to model brain function.

The Storm framework has minimal overhead so as to not impede the development of a diverse set of functional modules. The framework does, however, strictly enforce that any data shared between function modules must be contained within state variables: designers must be explicit and consistent in the organization of data into state variables.

Modeling the timing of function modules and state variables is an important aspect of the framework and is straightforward to use and experiment with. This allows a designer to focus on implementing behaviors and not be concerned with the implementation of timing constraints.

One possible disadvantage of using the framework is the strict enforcement on the organization of data into state variables. Experimental architectures may not want to make strong commitments to the separation of data; algorithms achieving high-performance may also require a high level of abstraction as realized in function modules and state variables.

In the future we plan to begin testing Storm's ability to scale by building iteratively larger and more complex architectures, as well as developing psychologically plausible models using state variables that map to brain regions and model brain function.

References

Alexander, A., Arbid, M. & Weitzenfeld, A. (1999). Web Simulation of Brain Models. *Proc. of the 1999 International Conference on Web-Based Modeling and Simulation*, 29-33. The Society for Computer Simulation International, San Diego, CA.

Anderson, J. R. (2007) *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.

Bothell, D. (2004). ACT-R 6.0 implementation.
http://act-r.psy.cmu.edu/actr6/

Eliasmith, C. Anderson, C. H. (2002). *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems.* MIT Press.

Fodor, J. A. (1983). *Modularity of Mind: An Essay on Faculty Psychology*. Cambridge, Mass.: MIT Press

Gray, W. (2007). *Integrated Models of Cognitive Systems*, Oxford University Press.

Kieras, D. & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. Human-Computer Interaction., 12, 391-438.

Munakata, Y., & Johnson, M. H. (Eds.) (2006). *Processes of Change in Brain and Cognitive Development: Attention and Performance XXI.,* Oxford: Oxford University Press.

O'Reilly, R. and Munakata, Y. (2000) *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*, Cambridge, MIT Press

Simen, P., Polk, T. A., Lewis, R. L. & Freedman, E. (2004). A computational account of latency impairments in problem solving by Parkinson's patients. *Proceedings of ICCM 2004*, Pittsburgh.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (Eds), *Advances in Neural Information Processing Systems: Proc. of the 1995 Conference*, 1038-1044. MIT Press.

Weitzenfeld, A., Arbib, M., and Alexander, A. (2002), *The Neural Simulation Language: A System for Brain Modeling.* MIT Press.

PART III:

A Computational Unification of Cognitive Behavior and Emotion

Cognitive science and emotion researchers represent two different communities with theories that cover different aspects of human behavior. Much of cognitive science fails to take into account the existence of emotion, both in the behavior that is modeled and the frameworks for developing models (current cognitive architectures). On the other hand, much of emotion research treats the rest of cognition as a black box and does not explore deep interactions between cognition and emotion.

Cognitive science and emotion research have an opportunity to inform each other. For example, emotion research can provide insight into areas such as motivation, innate goals and drives, and reflexive behavior that are not covered by standard cognitive theories. Cognitive science, on the other hand, can provide theories for all of the processing that is interleaved with emotion, thus providing more comprehensive and detailed theories and grounding for hypotheses about the functional role of emotion. Furthermore, research on cognitive architecture can provide a framework for developing computational models of the integration of cognition and emotion: enhancing the precision of emotion research, explaining the time course of behavior, the interaction of task knowledge and more primitive drives, as well as providing testable predictions of behavior.

As it has developed over the last 10-15 years, emotion theory has developed a "cognitive hole", while cognitive theories have an "emotion hole" – the theories from one area bump up against theories from the other but have not made significant progress in integration. The opportunity exists to start filling in those holes.

In this paper, we present a theory of agency called NAFO (Newell's Abstract Functional Operations) and show how a set of emotion theories called appraisal theories naturally fills in missing pieces in NAFO. NAFO is a set of abstract functional operations that all agents must perform in order to generate behavior (Perceive, Encode, Attend, Comprehend, Tasking, Intend, Decode, Motor). While NAFO describes the abstract operations, it does not specify the source and types of data that these operations manipulate. Finally, we claim that appraisal theory provides exactly the required data, neatly filling in this hole (and, conversely, that NAFO fills in the operation hole missing from appraisal theories).

In section 2 we provide background on cognitive and emotion theories, with a focus on NAFO and the specific appraisal theory we are using. In section 3 we describe the unification of these. In section 4 we describe our model, and in section 5 we present results for that model. Section 6 describes future work and section 7 concludes.

Background

In this section we will describe NAFO, a theory of agency, and then present background on cognitive theories, particularly Soar, in terms of NAFO. We will then present background on emotion theories, and make the connection between NAFO and appraisal theories as complementary pieces of the cognition/emotion integration puzzle.

Cognitive Systems

NAFO: An abstract computational agency theory

A theory of agency is a theory of cognitive control. It describes the *abstract functional operations* that compose cognition and allow for intelligent behavior (Newell 1990). NAFO is our renaming of Allen Newell's (1990) (awkwardly named) PEACTIDM, which is a theory of agency that specifies a particular set of abstract functional operations. PEACTIDM stands for the set of eight abstract functional operations he hypothesized as being the building blocks of immediate behavior: Perceive, Encode, Attend, Comprehend, Tasking, Intend, Decode, and Motor. These functions are *abstract* because although many of them may often be primitive cognitive acts, they can require additional processing; the details of this processing are not specified by Newell's theory.

In the description of NAFO, we will use the following simple immediate choice response task example adapted from a task described by Newell. As will become clear in this paper, even a simple example like this can have an emotional component. Suppose a subject is faced with two lights and two buttons. The lights are both within the subject's fovea. The subject's task is to focus on a neutral point between the lights and to wait for a light to come on. When a light comes on, the subject must press the button corresponding to that light. The subject then gets feedback that the correct button was pressed by the light turning off in response to the press. The subject's reaction time will be measured by how long it takes him to turn off the light. *Perceive* is the reception of raw sensory inputs. In this case, the subject will perceive one of the lights turning on. *Encode* is the transformation of that raw sensory information into features that can be processed by the rest of cognition. In this example, a representation is created that indicates one light has come on. *Attend* is the act of attending to a stimulus element. In this case it is not an overt eye movement but is some type of covert attention that must select the lit light even though the light is already foveated. *Comprehend* is the act of transforming a stimulus into a task-specific representation (if necessary) and assimilating it into the agent's current understanding of the situation, such as classification or identification. In our example, this takes the form of two concrete cognitive operations. First the subject verifies that it was one of the two lights that has come on (that is, his attention was not drawn by some other stimulus), and then the subject must discriminate which of the two lights came on. *Tasking* is the act of setting the task (i.e., the goal). In our example, Tasking took place before the task began – that is, the subject is already poised, looking at the lights with a finger ready to press a button and knows what to do. It is via this function that Comprehend knows what to expect and Intend knows what operation to choose. Given the task and the comprehension of the stimulus, *Intend* chooses a response and creates a prediction about the outcome the will result from that response. In our case, Intend will decide to press one button or the other, and a prediction will be created that the light will turn off. *Decode* translates the response from Intend into a series of motor actions. *Motor* executes the action; in our example, the pressing of the button. We will stop the example here, for now, and return to it later when we discuss emotion.

Newell argued that the functions performed by NAFO are both necessary and sufficient for immediate behavior (i.e., behavior at very short timescales). That is, he defined NAFO in terms of what was required by the Soar architecture, and sufficiency was shown by describing a simple scheme for implementing NAFO in Soar. In general, however, NAFO may not be a unique decomposition; there could be other theories of agency that embody the functionality in a different set of operators. Thus, an alternative theory might, for example, result in a different time course, but functionally would be equivalent. Newell also did not argue for the necessity and sufficiency of these functions across all general cognitive architectures, but we hypothesize that to be true.

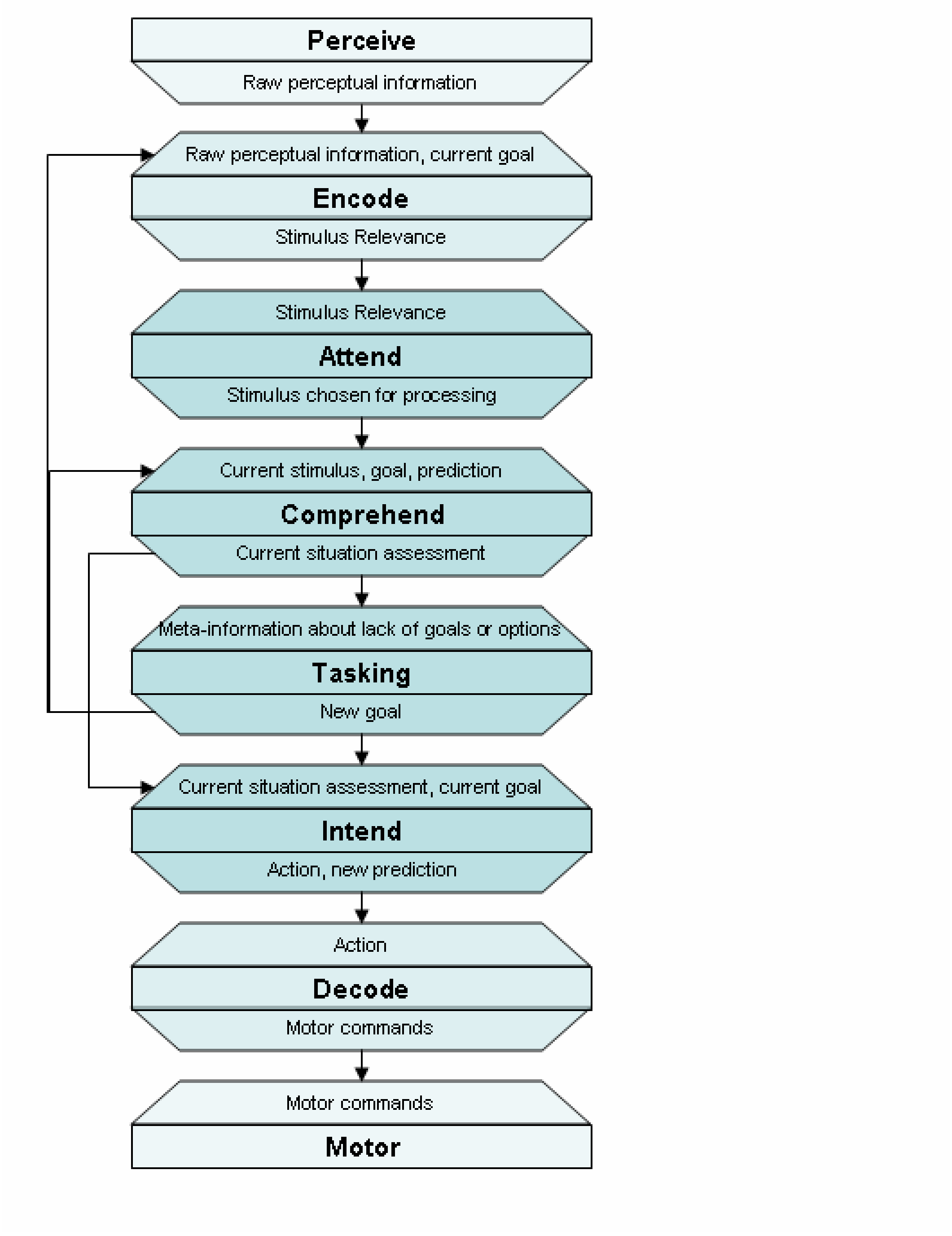

Perceive
Raw perceptual information
Raw perceptual information, current goal
Encode
Stimulus Relevance
Stimulus Relevance
Attend
Stimulus chosen for processing
Current stimulus, goal, prediction
Comprehend
Current situation assessment
Meta-information about lack of goals or options
Tasking
New goal
Current situation assessment, current goal
Intend
Action, new prediction
Action
Decode
Motor commands
Motor commands
Motor

Figure 24. Basic NAFO cycle

Newell argued that the ordering of these functions is largely predetermined by the data dependencies between the functions (see Figure 24). Perceive must occur before Encode, which must occur before Comprehend, which must occur before Intend which must occur before Decode which must occur before Motor. In some simple cases, the presence of a stimulus is all that is required for the task, and thus the Encoding step may be skipped. Tasking is the most flexible. It may occur between Comprehend and Intend (as presented). In this case the agent is deciding what task to attempt based on its comprehension of the situation. However, in many cases, the task may be known ahead of time and does not change during performance, in which case Tasking occurs first, as it did in the above example. However, Tasking must occur before Intend; otherwise, Intend does not have a context in which to take action.

Approaches to cognitive modeling

Although NAFO describes a set of abstract operations, it does not describe what mechanisms can realize these operations. There are different approaches to cognitive modeling that suggest different mechanisms.
One approach to cognitive modeling is called *cognitive modules*. For each individual functional capability, such as language or planning, there is an independent module. These modules then communicate in order to solve problems. In terms of NAFO, one can imagine having a separate module for each operation (in essence, making them concrete instead of abstract).

In contrast, the *cognitive architecture* approach decomposes cognition into computational components that are the building blocks for functional capabilities. Additionally, the interactions among these components give rise to temporal dynamics within the system.

A typical cognitive architecture consists of memories (both long-term and short-term) with different performance characteristics. For example, memories can differ what type of knowledge is stored/learned, and how knowledge is represented in the memory, and how it is retrieved. There can also be processing components that combine knowledge, such as to select between alternative interpretations or intentions. Most cognitive architectures also have perceptual and motor systems at some level. Many cognitive architectures also include features like attention and access to data about the operation of the system (meta-data); such if a memory fails to retrieve any items.

A core axiom of cognitive architectures is that they support generality and can be used across multiple tasks. It is the knowledge that is contained in their memories that is task-specific. Thus, an architecture's ability to solve any problem derives from its general subsystems, but its ability to solve a particular problem is derived from how its knowledge directs those subsystems to interact. Cognitive architectures are essentially frameworks for encoding and using knowledge.

In terms of NAFO, a cognitive architecture would implement the abstract operations via a combination of its subsystems and knowledge that directs those subsystems to interact in a way that enacts the operations. Given our expertise, we have chosen a particular cognitive architecture, Soar, in which to realize NAFO.

Soar

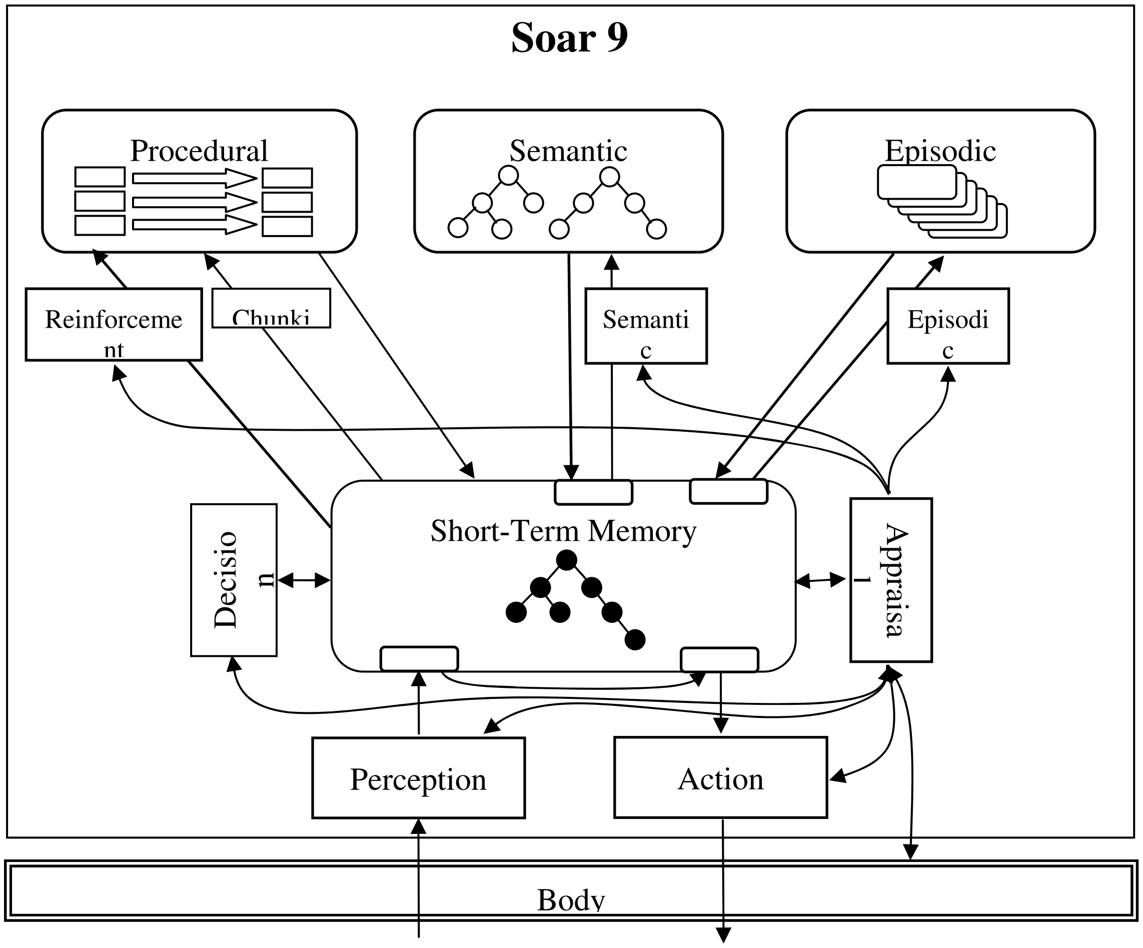


Figure 25. The Structure of Soar

Soar is a cognitive architecture that has been used both for cognitive modeling and for developing real-world application of knowledge-rich intelligent systems. Figure 25 is an abstract block diagram of Soar, which shows the major memories (rounded edges) and processing modules (square edges). In the bottom middle is Soar's short-term memory (often called its working memory). The short-term memory holds the agent's assessment of the current situation, derived from perception (lower middle) and via retrieval of knowledge from its long-term memories. It has three long-term memories: procedural (production rules), semantic, and episodic, as well as associate learning mechanisms.

Soar avoids the shortcomings of traditional rule-based systems by focusing deliberation on the selection and application of *operators*, which are explicitly represented in working memory instead of the focusing on the selection and firing of rules. Thus, rules, firing in parallel, propose, select, and apply operators, allowing the dynamic incorporation of knowledge to implement sub-steps in deliberation, providing more flexibility, and fine-grain representation of procedural knowledge than is possible in traditional rule-based systems.

Soar's general machinery directly supports NAFO. Perception results in new structures in short-term memory. Encode is realized as production rules that transform the structures in short-term memory into abstract symbols that can be recognized by other rules in production memory. Attend is implemented as an operator: that is, the agent chooses what to attend to. Comprehend is implemented as one or more operators that relate existing structures and symbols and create new structures and symbols in short-term memory to help the agent build an understanding of the ongoing situation. Tasking, or the management of goals, is realized via one or more operators that create new goal symbols and structures in short-term memory, and store and retrieve interrupted goals in semantic memory. Intend is one or more operators that choose motor actions and create predictions about the results of those actions. Decode and Motor are not directly implemented in Soar; rather, these are handled via an external mechanism that Soar interfaces to.

What NAFO and cognitive architectures provide

NAFO provides constraints on the structure of processing that are more abstract than cognitive architectures like Soar or ACT-R. While Soar and ACT-R specify processing units, storage systems, data representations, and the timing of various mechanisms, they are just building blocks and by themselves do not specify how behavior is organized to produce immediate behavior. NAFO specifies the abstract functions that these components must perform in order to produce intelligent immediate behavior.
Some of the key constraints that arise from the combination of NAFO and cognitive architectures are:

- The set of computational primitives that behavior must arise from (Cognitive architecture)
- The temporal dynamics of cognitive processing and behavior (Cognitive architecture & NAFO)
- The existence of core knowledge and structures that must be reused on all tasks (Cognitive architecture & NAFO)

Emotion Modeling

What can emotion provide?

NAFO and cognitive architectures describe processes and constraints on representation and the timescale of those processes, but they do not describe the details – it is up to the modeler to describe those, and the latitude for doing so is large. For example, where do goals come from, and how are they processed? What are the structures and knowledge that are used for processing? Consider PEACIDM: What structures does Encode generate? Given multiple stimuli, how does Attend choose what to focus on? What information does Comprehend generate? What information does Intend use to generate a response? One potential source for these required constraints is emotions.

We propose that much of the information required by NAFO is generated by the same processes that generate emotion, and that these processes are, in fact, the NAFO operations themselves. The abstract functions of NAFO need information about relevance, goals, expectations, and so on, and will be computing them in order to carry out their functions. The results of these computations, then, cause an emotional response.

Introduction to appraisal theories

Let us return to our previous immediate choice response task example. Suppose the task were modified in the following way: behind the scenes, the researcher can disable the button so the light will stay turned on even when the correct button is pressed.

When the button is functioning properly, things will proceed as in the unmodified task. When the subject presses the button, he will Encode the state of the light and Attend to it. In the Comprehend stage, he will verify that the light's state matches his prediction.

Suppose the first several trials go as just described. Then, the researcher disables the button. Since this has never happened before, when the subject Intends pressing the button, he will still create the same prediction – that the light will turn off. When the subject presses the button, though, the light does not turn off. Thus, when the subject gets to the verify step, a mismatch between the actual state and the expected state will be detected.

What is this mismatch? For this to be useful, the agent must actually generate a structure that represents the mismatch. We call this the Discrepancy from Expectation. This information allows the subject to react appropriately. But in order to react appropriately, the subject must have some notion of how likely the outcome was in the first place. That is, if Discrepancy is high, that does not have the same meaning if the subject's confidence in his prediction is low versus high. Thus, when the subject generates the prediction, he must also generate an associated Outcome Probability. In this case, since the subject had no reason to suspect that the light would not turn off, so we can assume his Outcome Probability was very high.

Since the Discrepancy from Expectation is at strong odds with the Outcome Probability, we expect the subject would experience surprise. The subject may not even believe what just occurred, and try to press the button again, going through the same steps. However, this time the Outcome Probability is probably much lower, and certainly after a few tries, the subject will realize that the button is not functioning. Emotionally, the subject's reaction may vary based on many factors, such as who he thinks is at fault (which we call the Causal Agent). If he thinks he broke it, he might feel shame. If he realizes he's being thwarted by the researcher, he might feel anger (especially if there was supposed to be some reward based on his performance).

The idea that there is a relationship between the way someone interprets a situation (along certain dimensions, such as Discrepancy, Outcome Probability, and Causal Agency) and the emotional response he has is described by a set of theories called appraisal theories, which is the approach we have chosen for our model. Appraisal theories argue that emotions result from the evaluation of the relationship between goals and situations along specific dimensions (see Roseman & Smith

2001 for an overview). Smith & Lazarus (1990) argued that, in general, emotions allow for a decoupling between stimulus and response, which is required for to allow organisms to adapt to a broader range of situations. This decoupling, then, meant that more complex cognition was required to fill in the gap. In other words, complex cognition goes hand-in-hand with complex emotion. Thus, one of the primary functions of this more complex cognition, we argue, is to support appraisal generation. Appraisal theories are appealing to us, then, because they are naturally described at the cognitive level, as opposed to the neurological or sociological levels. Thus, they should be a good match for cognitive architectures.

Appraisal theories are complementary to the general cognitive model we described. Whereas our cognitive model has functions and mechanisms but lacks a specification of the data and structures to be processed, appraisal theories primarily describe data (the appraisals) without providing a detailed description of the processes that lead to their creation and use. Attempts at describing the processes that generate and manipulate appraisals have been largely vague and under-constrained. Computational models (Gratch & Marsella 2004) show that processes can be described in detail, but there is no constraint on how the processes function or combine – these are degrees of freedom in current computational models of emotion that result in arbitrary implementation decisions.

These issues lead to many questions. For example, where do appraisals come from? Is there an independent process dedicated to producing appraisals or are they somehow tied into cognitive processing for a task? While many theories (from abstract to computational) attempt to address this in terms of various kinds of cognitive processing, important questions remain unaddressed. What causes the system to generate these values? Furthermore, why is one particular set of appraisals generated as opposed to another? In terms of processing, are there sequencing constraints such that some emotions are generated before others (Scherer 2001) or not (as most theories tend to believe)?

While existing computational models must address these issues to some degree in order to function, the deep theoretical integration with cognition is still missing. Integration with cognitive models can begin to address many of these issues by providing the mechanisms and processes that lead to appraisals and which utilize the results of appraisal (i.e., emotions, moods and feelings).

Scherer's appraisal theory

Just as we have chosen to implement our model in a specific cognitive architecture, Soar, we have also chosen a specific appraisal theory to work with: that proposed by Scherer (2001). We do not have a strong theoretical commitment to Scherer model, and we have chosen it largely because of the completeness of the theory (most appraisal theories have six to eight appraisal dimensions, while Scherer's theory has sixteen), which in some sense makes it the most challenging model to use. It is also complements our ideas in other ways, which will become clear when we describe the full model.

Scherer defines sixteen appraisal dimensions, as shown in Table 3. These dimensions are divided into four groups: relevance, implication, coping potential and normative significance.

Table 3. A mapping from appraisal dimensions to modal emotions with dimensions grouped by function (adapted from Scherer 2001). Those dimensions in italics are not implemented in our

current model. Open cells mean all values allowed. Abbreviations: Unfamiliar = Unfamiliarity, Unpredict = Unpredictable, Conducive = Conduciveness, intent = intentional, neg = negligence.

	Enjoyment/ Happiness	Elation/ Joy	Displeasure/ Disgust	Contempt/ Scorn	Sadness/ Dejection	Despair	Anxiety/ Worry
Relevance							
Novelty							
Suddenness [0, 1]	low	high/ medium			low	high	low
Unfamiliar			high		high	very high	
Unpredict [0, 1]	medium	high	high			high	
Intrinsic Pleasantness [-1, 1]	high		very low				
Goal Relevance [0, 1]	medium	high	low	low	high	high	medium
Implication							
Cause: Agent				other		other/ nature	other/ nature
Cause: Motive	intent	chance/ intent		intent	chance/ neg	chance/ neg	
Outcome Probability [0, 1]	very high	very high	very high	high	very high	very high	medium
Discrepancy from Expectation [0, 1]	low					high	
Conducive [-1, 1]	high	very high			low	low	low
Urgency	very low	low	medium	low	low	high	medium
Coping potential							
Control [-1, 1]				high	very low	very low	
Power [-1, 1]				low	very low	very low	low
Adjustment	high	medium		high	medium	very low	medium
Normative significance							
Internal Standards Compatibility				very low			
External Standards Compatibility				very low			

	Fear	Irritation/ Cold anger	Rage/ Hot anger	Boredom/ Indifference	Shame	Guilt	Pride
Relevance							
Novelty							
Suddenness [0, 1]	high	low	high	very low	low		
Unfamiliar	high		high	low			
Unpredict [0, 1]	high	medium	high	very low			
Intrinsic Pleasantness [-1, 1]	low						
Goal Relevance [0, 1]	high	medium	high	low	high	high	high
Implication							
Cause: Agent	other/ natural		other		self	self	self
Cause: Motive		intent/ neg	intent		intent/ neg	intent	intent
Outcome Probability [0, 1]	high	very high	very high	very high	very high	very high	very high
Discrepancy from Expectation [0, 1]	high		high	low			
Conducive [-1, 1]	low	low	low			high	high
Urgency	very high	medium	high	low	high	medium	low
Coping potential							
Control [-1, 1]		high	high	medium			
Power [-1, 1]	very low	medium	high	medium			
Adjustment	low	high	high	high	medium	medium	high
Normative significance							
Internal Standards Compatibility					very low	very low	very high
External Standards Compatibility		low	low			very low	high

Scherer's model differs from many appraisal theories in that it assumes a continuous space of emotion as opposed to categorical emotions. That is, many of the appraisal dimensions in the theory have continuous values, and Scherer proposes that each unique set of values corresponds to a unique emotional experience. Like all appraisal theories, Scherer provides a mapping from appraisal values to emotion labels, but he describes these labels as *modal* emotions – that is, common parts of the emotion space.

Another way in which Scherer's theory differs from most is that he proposes that appraisals are not generated at the same time. Rather, he claims that appraisals are generated in the order of the groupings given above for efficiency reasons. For example, there is no sense in wasting resources on computing the implications of a stimulus if the stimulus is irrelevant. We will return to this point after we have described our specific model.

Scherer also proposes a process model describing how, at an abstract level, the appraisals are generated and how they influence other systems, but it does not provide details of all the data needed to compute the appraisals, nor the details of those computations. Our computational model describes the details. Since the computational details include new constraints on how the model as a whole works, our model differs in some ways from Scherer's theory. This arises in part because of the need to develop a computational model of generation, and also because of the more limited scope of our model. Scherer's theory pays some attention to the physiological and neurological aspects of emotion, but like most appraisal theories, does not include detailed mappings from the theory to specific behavioral data or brain structures. Similarly, our model does not include a detailed physiological or neurological model, and does not yet attempt to model influences on cognition or action tendencies. Our model's primary focus will be on the generation of appraisals and how this leads to emotions, moods, and feelings, which in turn impact behavior.

One reason we find Scherer's theory so appealing is its completeness. Most appraisal theories only specify a six to eight dimensions but claim that there are probably others. Scherer is certainly open to there being even more dimensions than what he has specified, but the fact that he has specified so many gives us a better starting point.

Theory of integration

As we alluded earlier, we propose that the appraisal information is required by NAFO and that this information is computed by the NAFO operations themselves. The generation of these appraisals, and their accompanying emotional responses, then, occurs in the normal course of operations.
We will be using a subset of Scherer's (2001) appraisal theory (Table 3) as the basis for the rest of this discussion, but it should be possible to apply other comprehensive appraisal theories in a similar way. In the remainder of this section we will give a high-level overview of the integration. The next section will give a very detailed description of the model in a simple domain.

Let's consider an example that is more complex than our earlier example. Suppose a baseball player is faced with two competing stimuli: one is a fan waving at him from the stands, while the other is the ball, which is coming his way. Which stimulus should he attend to?

Once these stimuli have been perceived, the first thing the ball player needs to do is attend to one of them for further processing. The Encode function will abstract the stimuli in a way that allows Attend to determine which one to process next. That is, the structure generated by Encode should allow the agent to infer the *relevance* of the stimulus. There are several appraisal dimensions that describe the relevance of a stimulus, namely Suddenness, Unfamiliarity, Unpredictability, Intrinsic Pleasantness, and Goal Relevance. Thus, Encoding, by its very nature, must generate these appraisals so that Attend can do its processing. In this example, both the fan and the ball are Sudden and Unpredictable, but the ball is much more Goal Relevant (assuming his goal is to catch balls that come his way). Thus, Attend will use this relevance information to determine that the ball stimulus should be processed next.

Next, the ball player must determine what action to take in response to the stimulus he is processing, namely the ball; this is accomplished by the Intend function. The ball player must take several pieces of information into account when determining what to Intend. For example, where did the ball come from (e.g., was it the batter, or did the ball come from the stands?) He also needs to know his ability to deal with the ball is (e.g., does he have the Power to catch the ball, or is he going to miss it?) The player also has a prediction about the ball's trajectory; if this prediction is inaccurate (e.g., the ball takes a funny bounce), then the ball player may need to make adjustments. Again, this kind of information is described by several appraisal dimensions. In general, the Causal Agency appraisal identifies the source of the stimulus, while the Goal Conduciveness and Internal/External Standards Compatibility appraisal identifies the stimulus as something that is good or bad for the agent. Power, Control and Adjustment Potential appraisals help the agent assess its ability to respond, and Urgency identifies the timescale on which a response is required. The Causal Motive appraisal helps determine the kind of response that is required (e.g., an attack versus a reprimand). The Discrepancy from Expectation appraisal may help the agent determine whether planned actions are still applicable, while the Conduciveness appraisal indicates whether an action is needed to fix a bad situation or maintain a good situation.

Since this information is used by Intend, it must be generated by Comprehend. Comprehend derives this information by comparing the attended stimulus to the current task and prediction. It has at its disposal the full range of memories and mechanisms of a cognitive architecture to assist it as well. Thus, our ball player can compare the Encoded structure for the ball stimulus to his goal to determine that Conduciveness is high (since the ball is coming right at him) and to the current prediction to determine that Discrepancy from Expectation is low (since the ball is traveling along the predicted trajectory).

Tasking does not generate appraisal information, but it can be influenced by it. People often cope with their feelings by changing their goals (Gross & John 2003). Thus, if our ball player trips while trying to catch a ball, this can result in several appraisals characterizing the negative nature of the situation (e.g., low Conduciveness, high Discrepancy from Expectation, etc). This may cause the player to form a subtask to fix the situation (e.g., get up) and result in different behavior (e.g., the player must now run to catch the ball). The player may even decide to discard the task altogether (e.g., if he cannot possibly make it to the ball in time, or another player is now better positioned to make the play). These goal/task manipulation operations are examples of Tasking.

Figure 26 shows how appraisal information is generated and used by various steps in the NAFO process.

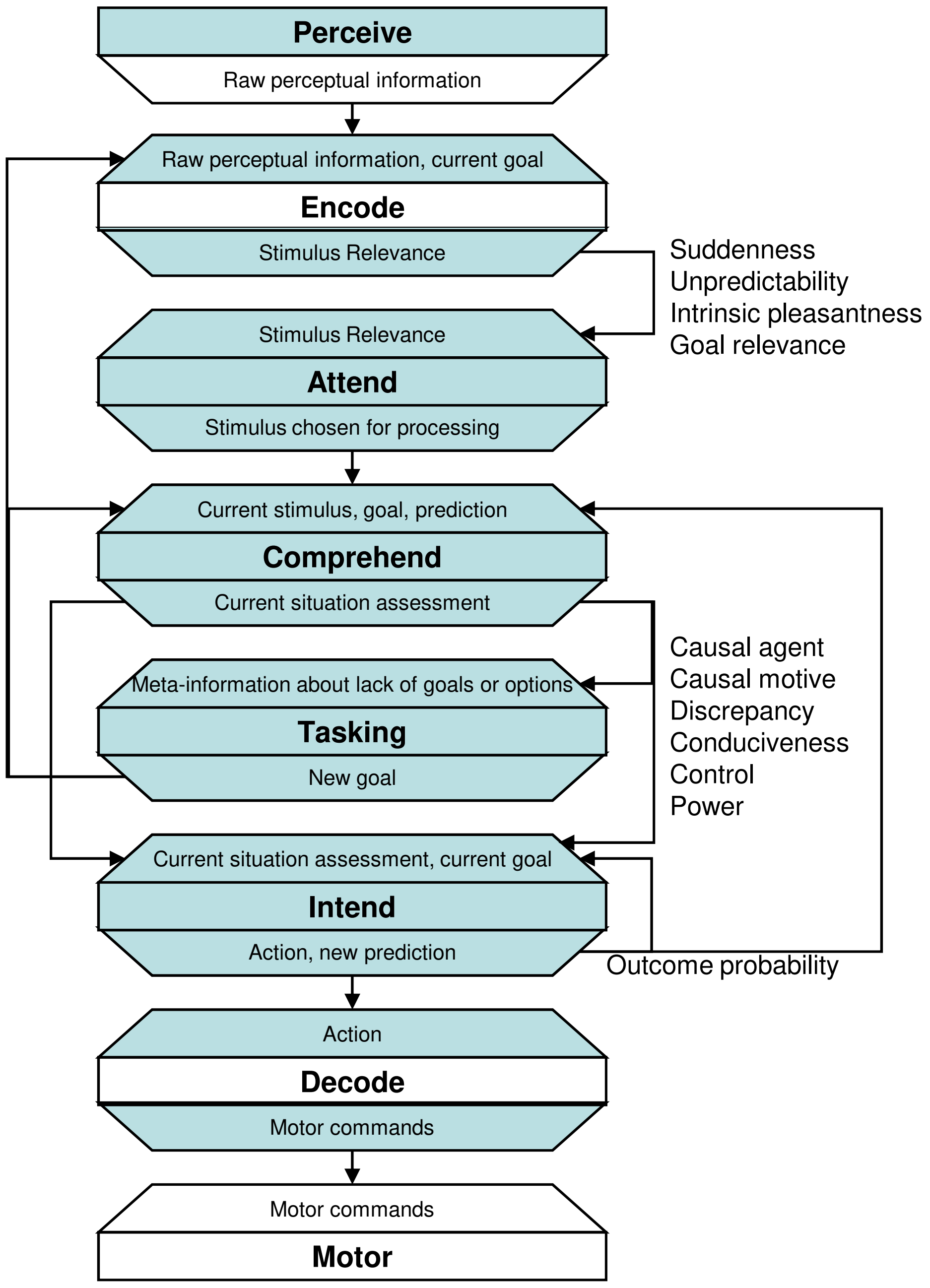


Figure 26. NAFO and appraisal. Appraisals define many of the inputs and outputs of various steps of the NAFO process. Encode generates various relevance appraisals, which are used by Attend. Comprehend generates various assessment appraisals which are used by Tasking and Intend. Intend generates the Outcome Probability appraisal, which is used by Comprehend, Tasking, and Intend.

The Model

Given the proposed theoretical overlap between cognitive architecture and appraisal theory, the next step is to attempt a computational integration. Our immediate goal is not to develop a high-fidelity model of human performance, matching detailed reaction times and error rates. Instead, our goal is to explore how the processing alluded to in the previous section can be realized at the level of detail required in a computational model and ask a few important qualitative questions. In our model, can we develop an integration that includes both appraisal theory and NAFO, while producing coherent, useful behavior? Do the appraisals arise from NAFO, or are new, additional phases of processing required to generate appraisals? Do the appraisals affect behavior and vice versa? Do appraisals have a reasonable (if not human-matching) time course?

Our model is implemented in the Soar cognitive architecture (Newell, 1990). Instead of using a standard psychological task for our exploration, we have chosen a simple Pacman-like domain called Eaters (Figure 27) that eliminates complexities of real-world perception and motor actions, while supporting tasks that although simple, can still allow for a range of appraisals and emotions. Eaters are a 2-D grid world in which the agent can move from square to square. The world contains walls that the agent cannot move through. The agent can sense the contents of the cells immediately to its north, south, east and west. The agent's task is to move from its starting location to a specified goal location. This may not always be possible, in which case an intelligent agent should probably choose to give up so it can move on to other tasks (i.e., so it can make progress in its life as a whole). The task ends when the agent notices it has achieved the goal (more on this later), or when it gives up.

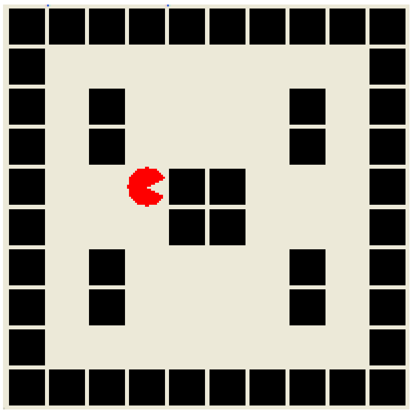


Figure 27. A screenshot of Eaters. The agent is the Pacman-like figure at location (3,4), walls are black cells, and open spaces are light-colored cells.

In terms of NAFO, the agent will need to Perceive its surroundings, including information about what lies in each direction (e.g., walls, open spaces), create structures representing the encoded form of the input (e.g., what is notable about each direction), Attend to one of the encoded structures, Comprehend that structure in terms of its current understanding of the situation (e.g., is it in line with predictions, what are the implications for the agent), Intend an action if possible (e.g., if the attended structure can be acted upon to get closer to the goal), and then perform the intended action (via Decode and Motor). Tasking will play a role when the agent is stuck; for example, it may need to create a subtask to circumvent a wall, or to give up.

In appraisal theory terms, each choice point (e.g., what to Attend to, what to Intend, when to give up) will be guided by emotional information. Thus, the steps preceding these choice points must generate the appraisals that, directly or indirectly influence the choices to be made.

What follows are the details of how each NAFO function is implemented in our model, including how the appraisals fit in.

Implementation of the NAFO Phases

This section describes how the NAFO operations are implemented in the model.

Perception and Encoding

We do not directly model the Perceive function. The Eaters domain provides symbolic inputs to the Soar agent. The Encode function is implemented as a set of elaboration rules that fire in parallel. These rules are capable of only simple monotonic inferences, and thus do not perform complex reasoning. They transform the input into a form that could be used for a wide variety of tasks. Each direction (north, south, east and west) is considered a stimulus that the agent can notice. A separate structure is Encoded for each stimulus. Each encoded structure includes information such as whether the direction is passable, whether it is on the path to the goal, whether it is on a path directly away from the goal, the distance to the goal, and whether the agent is making progress. If the agent is at a goal location, it will have a separate encoded structure for the goal completion, which is treated as an internal stimulus. This encoded structure is fairly general – any task in which stimuli can be represented as blocking progress to the goal or not, and in which a distance to the goal can be represented, can be Encoded in this way.

Consider the situation shown in Figure 28, which we will as a running example throughout the rest of this section. In this case, the agent is trying to reach location (7,4) and has just come from the west. It will have four encoded structures, one for each cardinal direction. The north, south and west structures will be marked as passable, directly off the path (since those directions will increase the distance to the goal), and at a distance of 4 from the goal. The east structure will be marked as unpassable but directly on the path to the goal.

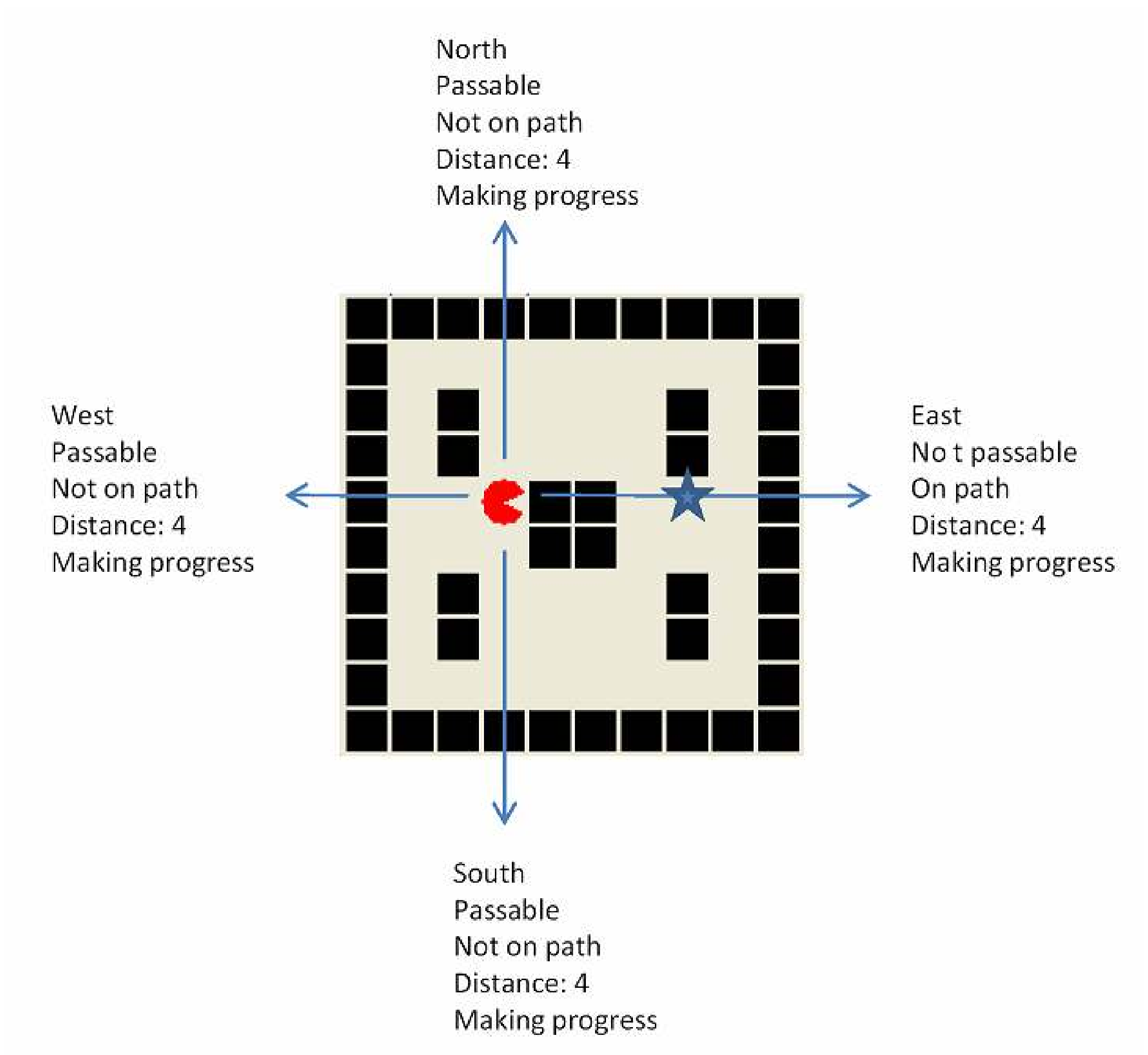


Figure 28. Encoded structures for each stimulus. The star shows the goal location.

Attending

Once the agent has these encoded structures, it must choose one to covertly Attend to for further processing. How can the agent choose? In general, the agent wants to make progress towards its goal, so stimuli that are Goal Relevant should be prioritized. However, Sudden or Unpredictable stimuli may also require attention, since these may be signals of danger or opportunity that needs to be dealt with. This is essentially an exploit versus explore tradeoff. Finally, some stimuli may be intrinsically pleasant or unpleasant (that is, independently of the current goal); such stimuli may also deserve attention. In our model, each stimulus is appraised along the Suddenness, Unpredictability, Intrinsic Pleasantness, and Goal Relevance dimensions. These appraisals contribute to an appraisal frame – the set of appraisals for a particular stimulus (Figure 29).

For the stimuli in our example, the north, south, and east stimuli have some Suddenness, whereas the west stimulus has no Suddenness (since the agent just came from there). In any environment, the agent will likely have some general expectations about what things to expect; our agent expects there to not be many walls in the world. Thus, the north, south and west stimuli have low Unpredictability, but the east stimulus has a high Unpredicatability. Our agent is also adverse to walls (since they only ever get in its way). Thus, it finds them Intrinsically Unpleasant. Thus, the east stimulus is has low Intrinsically Unpleasantness. Finally, since the east direction is on the path to the goal, it is highly Goal Relevant, but the other stimuli are not.

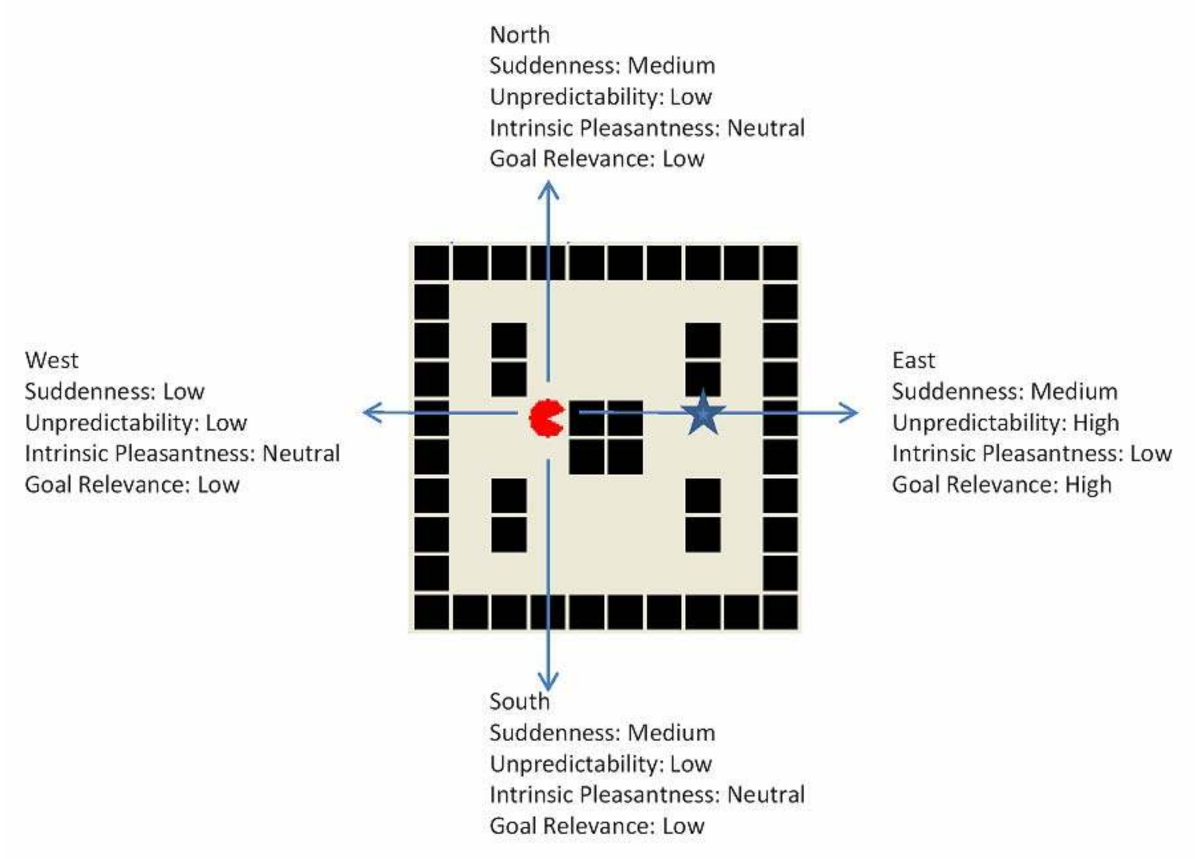


Figure 29. Pre-attentive appraisal frames for each encoded structure

The choice of which stimulus is Attended to is a weighted random choice, with weights determined by the values of the appraisals just discussed. Since unusual stimuli are more likely to be worthy of attention, as described above, appraisals with more extreme values lead to larger weights; that is, more interesting stimuli are more likely to be attended to. Thus, the appraisals have provided a task-independent language that allowed us to generate control knowledge.

In our example, the north and south Attend proposals have moderate weights, whereas the west Attend proposal has a slightly lower weight (since its Suddenness is lower). The east Attend proposal has a higher weight because it is on the path to the goal, leading to an appraisal of Goal Relevance, and it has a wall, which is Intrinsically Unpleasant. Thus, the agent is likely to Attend east.

Active Appraisal Frames

An appraisal frame is simply the collection of current appraisal values; that is, the set of appraisals that describe the current situation as the agent is thinking about it (Gratch & Marsella 2004). Before an agent Attends to a stimulus, there are several appraisal frames that have been started – one for each stimulus the agent is perceiving. We call these the pre-attentive appraisal frames. As soon as a stimulus is attended to, its appraisal frame becomes the active frame, which determines the current emotion. That is, there is a mechanism called the Appraisal Detector (Smith & Kirby 2001) that processes the active frame. It is via this mechanism that the active frame affects the rest of the system. Emotion theories disagree as to how many emotions a human can have at once. Our current model only supports one active appraisal frame at a time, and thus only one emotion (not to be confused with mood or feeling, which are separate; these will be discussed later). For pragmatic reasons, the pre-attentive appraisals generated for the other stimuli do not influence the current emotion in our model. Becoming active allows several other appraisals to take place. This is in line with our hypothesis that Comprehension follows Attend, and that Comprehension must generate the data necessary for further processing (e.g., Intending an action).

Comprehension

Next, the agent performs the Comprehend function, which adds several appraisals to the active frame. The agency of the stimulus is determined (in our example model, "nature" is always the Causal Agent and "chance" is always the Causal Motive). The Conduciveness is also determined – if the stimulus direction is passable and on the path to the goal, it has high Conduciveness, whereas if it is off the path or blocked it has low Conduciveness. The Control and Power appraisals are also generated – if a stimulus direction is passable, Control and Power are rated high, whereas if the direction is unpassable, Control and Power are low.

In our example, since the agent is Attending to the east stimulus, which is unpassable but on the path to the goal, it will generate appraisals of low Conduciveness, low Power, and low Control (since it can't walk through walls). Causal Agency and Motive are "nature" and "chance", as noted above.

Next, the agent compares the stimulus to the current prediction (as generated by Intend – see below) to determine whether they match. The concrete Comprehension operator that does this is called Verify, and leads to the generation of the Discrepancy from Expectation appraisal. If the stimulus is a match, then a Discrepancy from Expectation appraisal is low; if there is not a match, then the Discrepancy is high.

Once a stimulus has been verified, the agent must determine if further processing is warranted. Specifically, can additional processing of this stimulus lead to an action that helps the agent? The agent uses a heuristic called *dynamic difference reduction* to make this choice. Difference reduction (Newell, Shaw & Simon 1960) is the idea that an agent should take mental steps that reduce the difference between the current state description and the goal state description. Dynamic difference reduction (Agre 1988) is the idea that the agent shouldn't take such steps internally, but should actually take them in the world (to avoid the need for increasing amounts of memory to track one's imaginary progress). Thus, difference reduction leads to plans whereas dynamic difference reduction leads to actions. In our model, if a stimulus can be acted upon (i.e., it is associated with a passable direction) and it does not lead directly away from the goal, then Comprehension is complete and the agent acts upon it (it does the Intend function). Otherwise, the agent chooses a second Comprehend operator, Ignore. Ignore marks the stimulus as processed and allows control to return to Attend, which will choose another stimulus to process from the remaining stimuli as above. This deactivates the appraisal frame for the Ignored stimulus.

In our example, the agent is Attending east, which is a wall (Figure 30). The Verify operator will find a mismatch (since our simple model almost always predicts a passable route to the goal). This will trigger an appraisal of high Discrepancy from Expectation, which is added to the current frame. Since there is a wall, the agent cannot directly act upon the stimulus, so it then Ignores it. In fact, the agent is trapped by its goal in this case. As it Attends and Comprehends to each stimulus, it will find that the remaining stimuli lead away from the goal. Thus, the agent will eliminate all of its options.

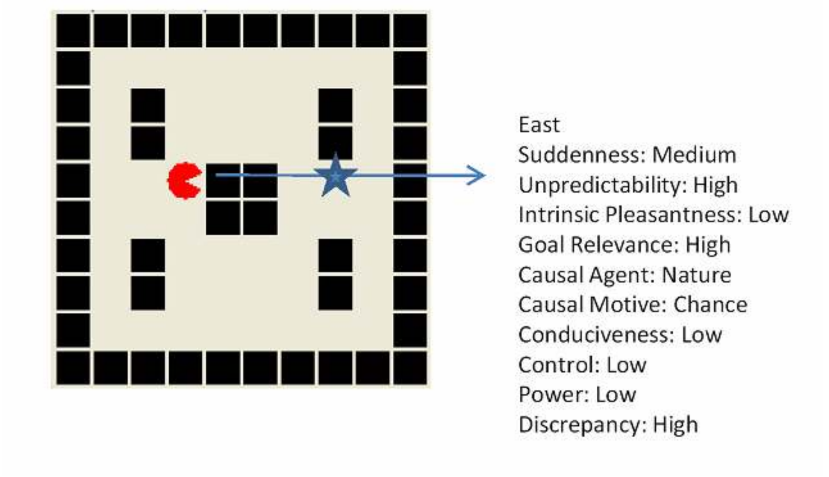


Figure 30. The agent attends East, making that appraisal frame active. The Comprehend function adds to this frame. The agent decides to Ignore this stimulus.

Tasking

When the agent has no options left, it is forced to engage in Tasking. Generally speaking, Tasking is about manipulating goals (e.g., creating goals, giving up on goals, etc.). In this case, the agent will form a subtask to get around the blockage (Figure 31). In general, there are at least two types of goals. One type is abstract – the goal cannot be acted upon directly and must be broken down into more concrete components (perhaps many times) until it is in a form that can be directly acted upon. For example, the goal "Go to Work" is very abstract, and must be broken down to something that can be directly executed, such as "take a step". The other type is concrete – the goal can be directly acted upon. This is the form of goals in our model. When the agent temporarily retasks itself for the purpose of making progress on its original goal, we call this subtasking, and we call the new goal structure a subtask.

The goal that the agent cannot make progress on is to go to (7, 4). The reason the agent is stuck on this goal is that its control knowledge and task formulation are too restrictive. Movement in any available direction would take it further from the goal, which violates its dynamic difference heuristic. In order to move around the blockage, it needs to temporarily get further away from the goal. Thus, the agent needs to retask to create a goal that is less constraining, allowing it to get further from the main goal, but without violating its constraints in the new goal. The agent does this by defining the step it would ideally take – in this case, it would ideally move east to x=4. It sets this as its new subtask. That is, there is no constraint in the y (north-south) direction.

When an agent creates a subtask, it records information that gives it some idea of whether it is making progress or not. Specifically, it records the distance to the parent task (goal) at that time. It also tracks the minimum distance it has ever been to the goal upon entering a subtask. If the current distance to the goal is less than the minimum distance to the goal, then the subtask is considered a "good" subtask – that is, the agent knows that, even though it has to retask, it is making progress towards the goal. If the distance to the goal is not reduced, then the subtask is considered a "bad" subtask – that is, the agent cannot tell if it is actually making progress by retasking. The Encode function notes whether the current subtask is good or bad on each encoded structure, and this information influences some of the appraisals. In this model, appraisals are more positive in good subtasks.

As alluded to above, once the agent has this new subtask, the encoded stimuli are regenerated (since there is a different context for them now) and the agent can then re-attend to the stimuli to see if any are now suitable. The agent can theoretically create an arbitrary number of nested subtasks this way, but for the current task it only needs one at a time (although it may create several in the course of completing the goal).

In our example, this is the agent's first subtask, so it defaults to a good subtask. The agent might still attend to the east stimulus first and ignore it again, but when it attends to, for example, the south stimulus, it will find that it is no longer directly off the path to the subtask. Instead it is now a sideways move (since it neither gets it closer to nor further away from x=4). Thus, the agent determines that this stimulus can be used for Intention processing.

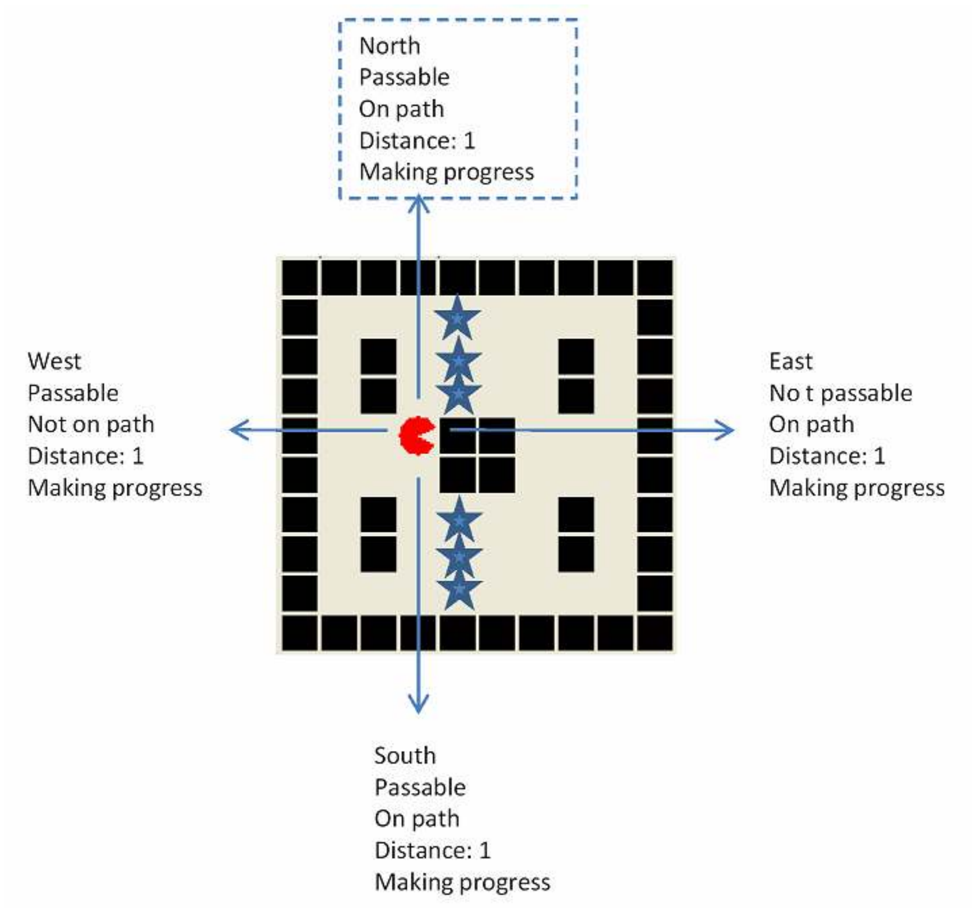


Figure 31. The agent creates a subtask to get around the blockage. The stars show the possible locations that would solve the subtask. This causes new encoded structures to be created. The agent Attends north.

Intending

Once the agent has found a stimulus it can act upon, it performs the Intend function, which is also implemented as a Soar operator. Intend proposes moving in the direction of the stimulus. It also creates a new prediction structure – namely that the next stimulus direction will be passable and on the path to the goal (in our model, the agent is always optimistic in this way). If the agent is currently one step away from the goal, then it creates a goal achievement prediction. Along with the prediction, the agent also generates an Outcome Probability appraisal. The Outcome Probability is tied to the prediction, and thus all appraisal frames in the situation that results from an Intend will inherit this same Outcome Probability (Figure 26).

In our example, Intends proposes moving north (Figure 32). The Intend operator sends a command to the environment to move north, and also creates a prediction. The prediction structure is simply a structure, with the same form as an encoded structure, that describes a stimulus that is passable and on the path to the subtask (as mentioned above, the agent is always optimistic in this way). Since it is pursuing a subtask, the agent is less confident of its predictions, so it only rates the Outcome Probability of this prediction as moderate.

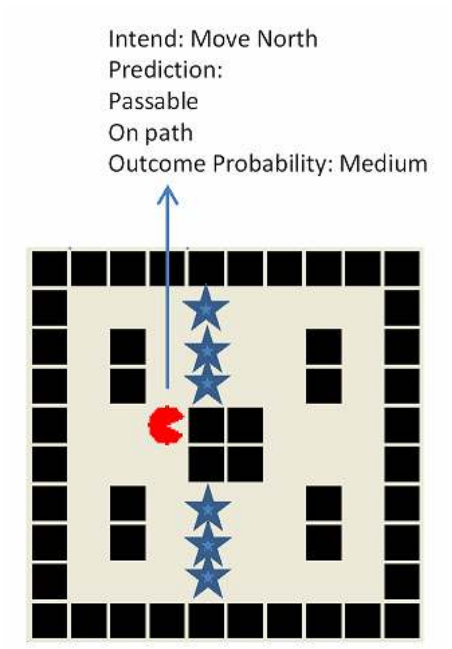


Figure 32. The agent Intends moving north. It creates a prediction of the next event it will see.

Decode and Motor

We do not directly model the Decode and Motor functions. The model uses Soar's standard method of communicating an action command to the simulated environment, which then executes it, leading to a new input state.

Other Implementation Details

This section describes other implementation details of the model that are not directly related to NAFO. That is, the details in this section are not about the functional needs of the agent as we have been describing them thus far, although we plan to show in future work that they are functional, for example, in learning.

Emotion, Mood, Feelings, and Feeling Intensity

Thus far, we have described how the agent creates appraisal frames and how, at the pre-attentive level, those appraisals affect the agent's behavior. We have not yet addressed what the model does with activated frames and how these affect the agent's behavior. In this section, we will describe how the agent uses appraisals to generate emotions, mood, and feelings.

A distinction not made by many theories is that between emotion, mood and feelings. One existing distinction made between emotion and mood is in terms of timescale: emotions are short-lived while moods tend to last longer (Rosenberg 1998). Many physiologically-oriented theories of emotion (James 1890, Damasio 1994, 2003) distinguish between emotions and feelings: emotions have some impact on physiology, and the agent perceives or *feels* these changes, called the agent's feelings. That is, feelings are our perception of our emotions.

In our model, we maintain a distinction between emotion and feeling, and also introduce mood. Emotion is the currently-active appraisal frame. In our model, we use a simple abstraction of mood which is essentially an average over recent emotions. In general, mood may be much more complex in both its antecedents and effects. Feeling, then, is the combination of emotion and mood, represented as an appraisal frame, augmented by an intensity. The intensity of the feeling is determined by two factors. The first factor is called the “surprise” factor, which is calculated from the Outcome Probability and Discrepancy from Expectation appraisals in the feeling frame. Basically, the surprise factor is high if the agents expectations do not match what actually happened and low if they do. The second factor is just an average of the other appraisal values in the feeling frame. Details of exactly how we calculate mood, feeling, and feeling intensity are reported in Marinier & Laird (2007).

Behavioral Influences of Emotions, Mood and Feelings

Up to this point, we have described a theory of the origins of emotion mood and feelings, but a critical question remains: how do mood and feelings influence behavior? Emotion theories describe a number of influences, including affects on cognitive processing (Forgas 1999), action tendencies (Frijda et al 1989) and coping (Gross & John 2003). Our current approach is very simple, included to demonstrate the possibility of feelings influencing behavior and focusing only on one aspect of coping: coping by giving up on goals.

Most AI systems, when faced with a difficult or impossible task, fail to recognize this and will work on the problem until all resources are exhausted. By providing emotional feedback, our model allows the agent to detect that it is not making progress towards the goal, and thus it can choose to discard that goal (possibly so it can move on to another goal or stop wasting resources). In our model, as long as the agent is making progress on its task, it will continue with that task. However, as described earlier, if it fails to make direct progress, it will form a subtask. While pursuing a subtask, the agent can choose to give up if its current feeling of Conduciveness is negative. Giving up is another form of Tasking – it removes the current goal. As this feeling intensity increases, the agent is exponentially more likely to give up. The option to give up is in competition with other activities in the subtask, specifically attending to possible spaces to which it can move. As the agent eliminates more of its attend options (by attending to and then ignoring them), it becomes more likely to give up (since there is less competition from other attend proposals).

While the current model only has this single direct influence of feelings on behavior, each appraisal of each stimulus has an indirect influence. As described above, at the Attend stage, the pre-attentive appraisals influence where attention is focused next. Furthermore, past appraisals influence the current feeling via mood, and thus indirectly influence the agent’s decision to give up or not.

Labeling Appraisal Frames

While the agent does not use linguistic labels to determine its behavior, we found such a labeling function is useful in analyzing the agent’s behavior (indeed, we will use it in the results reported below). The labeling function is based on the Manhattan distance between the agent’s appraisal frame and the modal emotions defined by Scherer. Since some modal emotions have many unspecified values (which are treated as distance 0), some emotions are frequently closer to the feeling frame than others, even when their specified appraisal values are not good matches. Elation/Joy is one such emotion (it has open values for Intrinsic Pleasantness, Discrepancy from

Expectation, Control and Power). To compensate for this, we only considered modal emotions that have a Conduciveness with the same sign (or an open Conduciveness). In other words, we divided the emotions into positive and negative emotions based on Conduciveness, and ensured that only labels with the same valence as the frame could be applied. Thus, it is not possible for a feeling with negative Conduciveness to be labeled as Elation/Joy.

Model Summary

To summarize, our model is based on NAFO and implemented in Soar. The structures that many of the NAFO steps use and generate are based on appraisals. The Encode step generates the Suddenness, Unpredictability, Intrinsic Pleasantness and Goal Relevance. It also generates encoded structures. The Attend step uses these to choose a stimulus to attend to. Attending also enables the generation of other appraisals, including Conduciveness, Causal Agent, Causal Motive, Control, and Power. The Comprehend step is implemented as a Verify operator (which generates Discrepancy from Expectation) and an Ignore operator. Tasking allows for the generation of subtasks when there are no useful actions to take. Intend takes the action associated with the currently-attended encoded structure and creates a prediction of the outcome of that action, as well as the Outcome Probability of it. When pursuing a subtask, the agent has the opportunity to give up (another Tasking operator). This combination of appraisal and NAFO leads naturally to sequential constraints on appraisal generation.

The probability of giving up is influenced directly by the feeling's Conduciveness dimension, and also indirectly by all the other numeric appraisals via the intensity of the current feeling. Feelings are determined by combining mood and emotion, with mood being influenced by emotion.
The model as presented do not use the Unfamiliarity, Urgency, Adjustment, Internal Standards, or External Standards appraisal dimensions from Scherer's theory. These were not critical for our example implementation and adding these to our architecture is future work.

Finally, we want to note that although the model is implemented in Soar using Scherer's appraisal theory, the underlying theory is intended to be general. That is, we have not intentionally introduced any constraints that would prevent this theory from being implemented, for example, in ACT-R using a different appraisal theory.

Implications of the Model

In this section, we discuss the following topics: the capabilities of the model, and the sequential generation of appraisals, and the non-categorical nature of feelings.

Capabilities

What kinds of tasks can this architecture model qualitatively? The architecture is fairly general: it allows for arbitrary perception and action, with the restriction that these can be encoded symbolically. It is task- and domain-independent: any task- or domain-specific information can be used in the Encoding, Comprehension, and Intend knowledge. It specifies a time course for emotion, mood, and feeling: coupled with parameters for the length of a decision cycle (assumed to be 50 milliseconds in Soar), this can also lead to both qualitative and quantitative predictions. While it does have limitations, such as not modeling the impact of emotion on the functioning of the underlying cognitive architecture, these limitations are not fundamental and could be addressed by future extensions.

Sequences of Appraisals

In Scherer's theory, he proposes that the appraisals are generated sequentially because the outcomes of some appraisals obviate the need for others. For example, if none of the initial appraisals (Suddenness, Unfamiliarity, Unpredictability, Intrinsic Pleasantness and Goal Relevance) indicates that something interesting is happening, then there is no need to continue processing the stimulus. Our model also imposes sequential constraints (see Fig), but for two reasons, one of which is related to Scherer's. Attend will not attend to a stimulus unless one of the above mentioned dimensions indicates that it is interesting, much like Scherer's theory describes. However, additional ordering constraints arise from the flow of data in the model. For example, since Discrepancy from Expectation arises from the Verify operator, which is part of the Comprehension function, it occurs after the Conduciveness appraisal (which is activated upon attending). Similarly, the Outcome Probability appraisal is generated in the Intend step, which comes after Comprehension. Thus, while Scherer's argument behind sequential appraisal generation centers on efficiency and the wastefulness of generating irrelevant appraisals, our data-driven model imposes an ordering because the appraisals cannot be generated earlier (regardless of the efficiency). Thus, in addition to incorporating Scherer's original efficiency concerns, we extend his theory by including data-driven constraints as well. The idea of appraisals being data-driven has been mentioned elsewhere (see Roseman & Smith 2001 p.12-13 for a brief overview of this point), but the idea has been used to argue that appraisal ordering is not fixed at all. Data-driven processing combined with NAFO implies at least a partial ordering.

Non-categorical feelings

In many systems (Neal Reilly 1996), what we call the "feeling" is reported as a label (such as anger, sadness, joy, …) with an intensity. These *categorical* theories of emotion assume that there are a small, fixed number of possible feelings that vary only in intensity. In our model, like in Scherer's (2001) theory that inspires it, each unique appraisal frame corresponds to a unique experience. Thus, in our model, because we have no model of physiology, the feeling input is the feeling appraisal frame, augmented with an intensity. Not only does this give us a nearly infinite space of possible feelings, but it also allows us to directly compare the emotion and the feeling, which helps us understand the model. This is not to say that we cannot generate categorical, linguistic labels for these frames. We can and do for our own analytical purposes, but the current model does not use these labels, and even if it did, at best such labels would only be a model of how an individual in a particular culture might label the feelings.

Results

In addition to the qualitative properties described above, some quantitative results can be provided. What we want to demonstrate is that:

1) The model works and produces coherent behavior.

2) Different environments lead to differences in behavior, including:
   a. Different time courses
   b. Different feeling profiles
3) Feelings impact behavior.
4) In a given environment where the agent has choices, these choices impact feelings and thus the agent's success.

Some of these properties were described earlier in the architecture design, but now we present results to show that the design was successful in producing these properties. For simplicity, we used a non-human agent in the simplified environment described earlier to demonstrate the architecture. Thus, while we present time course data, this data should not be mapped onto real time for comparison to humans.

We placed an agent in several different mazes in the Eaters domain with a specific goal location in each. In each maze, the distance from the start to the goal was 44 moves (except for the last maze, in which it was impossible to reach the goal). Our aim in designing these mazes was to place the agent in progressively more difficult situations to demonstrate the properties listed above. In the first maze (Figure 33), the agent did not have to ever retask to reach the goal, and there were no distracting stimuli; that is, it could not see any walls on its way to the goal. The second maze (Figure 34) is exactly the same as the first except that the path to the goal is lined with walls (and hence distractions). The third maze (Figure 35) is very similar to the second, except that there is a kink in the path that requires a brief retasking to maneuver around. This is because the agent has no direct way of making progress when it reaches the kink – if it moves north, it will be further from the goal, and it can't move east because of the wall. Thus, retasking allows it to temporarily move further from its original goal. The fourth maze (Figure 36) contains twists and turns such that four subtasks are required to reach the goal. In the fifth maze (Figure 37), it is not possible to reach the goal.

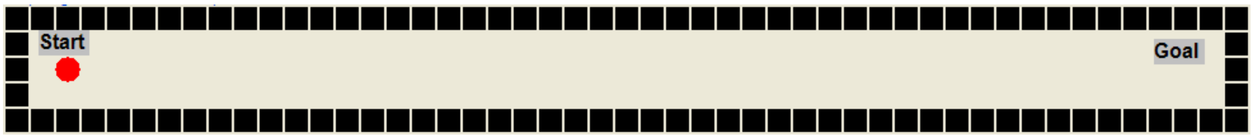


Figure 33. An Eaters maze without any distractions.

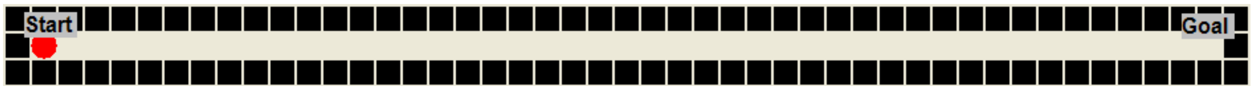


Figure 34. An Eaters maze with distractions.

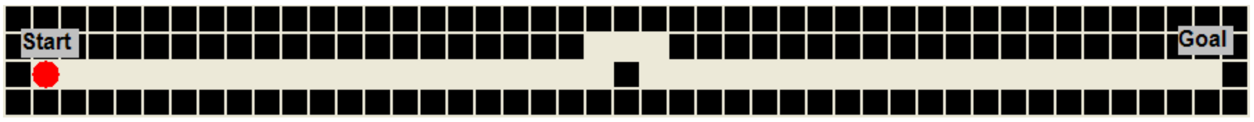


Figure 35. An Eaters maze with distractions and one subtask.

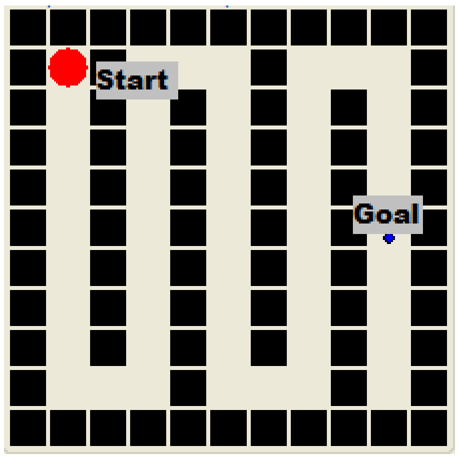


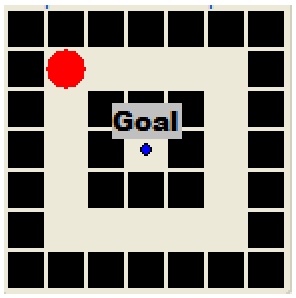
Figure 36. An Eaters maze with distractions and multiple subtasks
.


Figure 37. An Eaters maze that cannot be successfully completed.

In the first two mazes, the agent will never give up, since it never has to retask. However, we anticipate that the distractions from the walls in the second maze will make it take significantly longer to complete than the first, and that the agent will experience more negative emotions as a result. In the last three mazes, retasking is required and thus the agent can fail. In the third and fourth mazes, the addition of the subtasks should require extra processing that causes the agent to take longer to complete the mazes. In fact, the agent may often give up before completing the fourth maze. The agent will always give up on the fifth maze because it is not possible to solve. We expect this to take less time than the fourth maze, because in the fourth maze the agent is always making progress, whereas in the fifth maze, after the first subtask, the agent will detect that it is not making progress, which should lead the agent to feel worse and hence give up sooner.

Figure 38 shows the time course of behavior in the different mazes, as well as the success rate in each maze. As we predicted, the mazes do lead to different time courses, which fulfills property 2a (different time courses). In general, as the mazes increase in difficulty, the agent takes longer to complete (or give up on) them. When the agent does give up, though, it takes less time. This makes sense since the agent is stopping early. Still, the maze with multiple subtasks takes longer than the maze with a single subtask when the agent gives up. The impossible maze takes slightly less (but still statistically significant) time to give up. This is because, after the first subtask, all subtasks are considered "bad" subtasks, whereas in the other mazes all subtasks are "good" subtasks. This should mean that there are more negative appraisals in the impossible maze, causing the agent to feel worse and thus give up sooner.

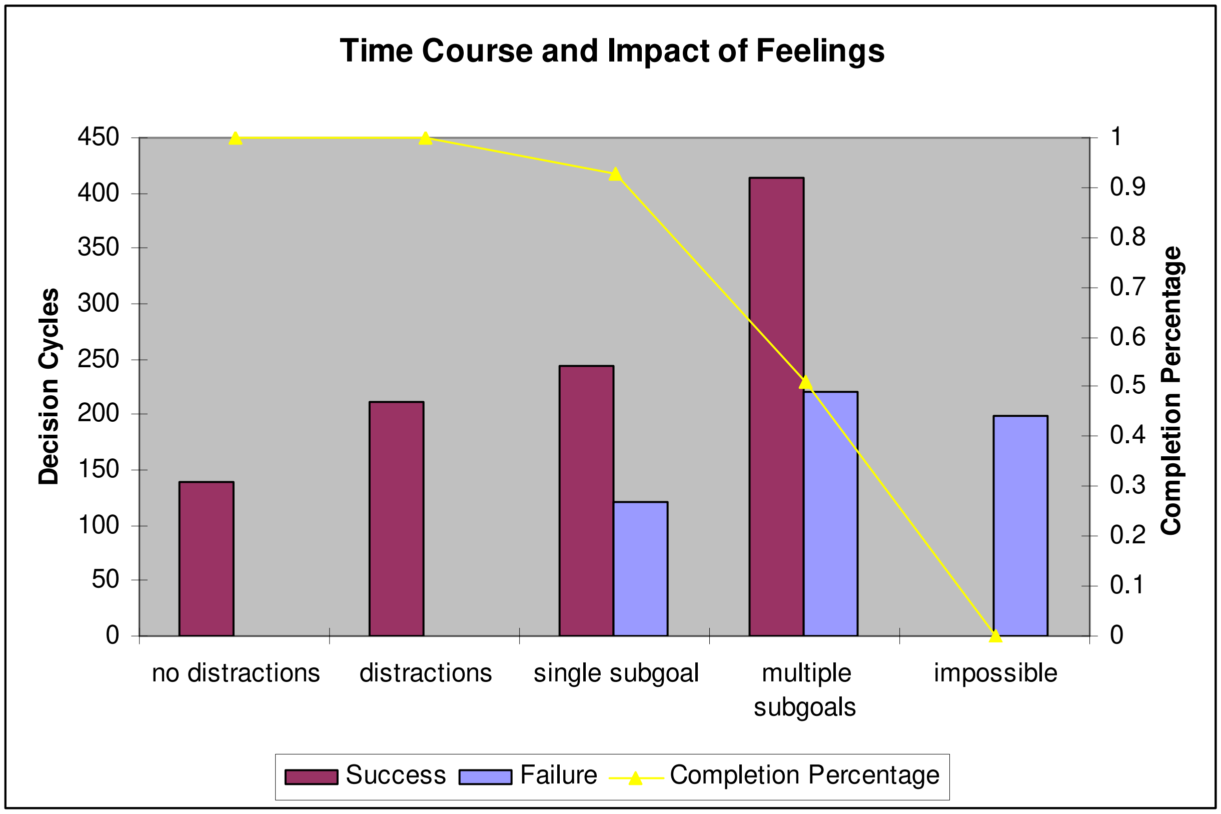


Figure 38. The bars show the number of decision cycles required to complete each maze for the success and failure cases. The line shows the success rate. All differences are statistically significant (1000 trials for each maze, >95% confidence level).

In Figure 39 we see that this is indeed the case. The feeling labels in the figure are generated as described in Section 0. In each maze's feeling profile, the positive feeling (elation-joy) instances outweigh the negative feeling (anxiety-worry and displeasure-disgust) instances except for the impossible maze, where the negative feelings dominate. We can also see that each maze produces a different feeling profile, and that feeling profiles also differ between the success and failure cases. This supports property 2b (different feeling profiles). In contrast, the failure cases for mazes 3 and 4, the positive and negative feelings are nearly equal. This is to be expected given that the subtasks are "good," the agent positively appraises every move it makes (since it thinks it is making progress). Thus, this offsets the negative feelings to some extent. However, each negative feeling in a subtask represents an opportunity to give up, and these more frequent opportunities lead to failure.

This, together with the data from Figure 38 supports properties 3 (feelings influence behavior) and 4 (choices influence feelings). That is, success and failure (both absolutely and in terms of rate) are defined by different feeling profiles, implying that feelings do influence behavior. Furthermore, even within the same maze the success and failure cases have different profiles, implying that the choices the agent makes in those mazes impacts feelings and behavior.

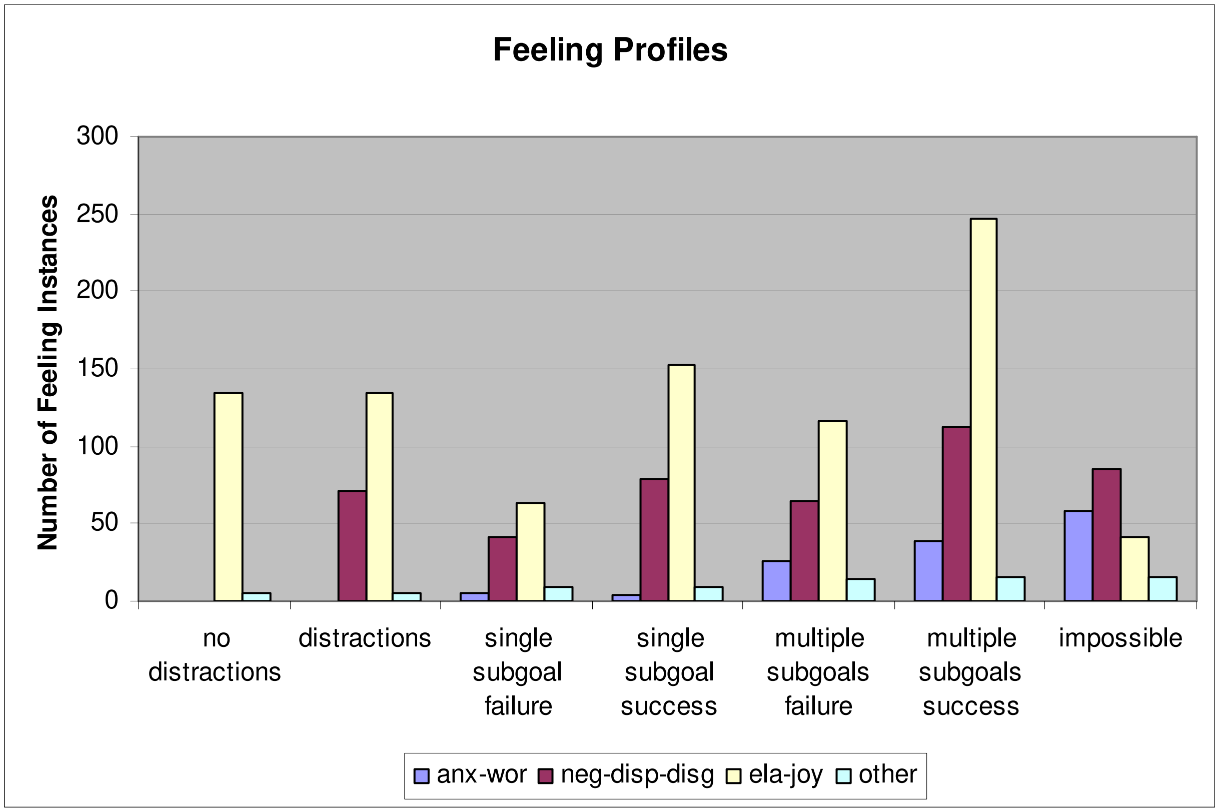


Figure 39. The number of times each kind of emotion (as labeled by our labeling function) occurred in each maze with the success and failures for mazes 3 and 4 reported separately. "Other" includes Boredom-Indifference, Fear, Positive Displeasure-Disgust, and Sadness-Dejection. Differences are statistically significant (1000 trials for each maze, >95% confidence level).

Finally, the above analysis supports property 1 (coherent behavior). That is, the agent's behavior and feeling profiles are expected given its task and environments.
As a final comment, as shown in Figure 39 the agent experiences a wide breadth of feeling types in these mazes (seven different kinds according to our labeling function). Given the limited nature of the domain, one might expect a much more limited set of feelings. Indeed, we have shown that multiple feelings can arise from simple manipulations of the environment, even in similar situations. One way is via interactions with the goal – adding structure that requires subtasks leads to many different feelings emerging. Another way is via interactions between mood (including decay) and emotion. Sometimes, even though we might classify a mood one way and an emotion another way, their combination results in yet another classification. This is an interesting prediction that could help explain why people are sometimes confused about their feelings.

Future Work

There is much left to explore in the integration of cognition and emotion. A major area that we have not addressed is the impact of emotion on the cognitive architecture's subsystems, such as memory and decision-making, and how this might impact areas such as complex problem solving. Furthermore, we have not explored major emotional phenomena such as action tendencies, the influence of physiological drives, and social and cultural interaction. Another major area for future work is in the relationship between emotion, cognition, and learning.

Additionally, our test domain is highly artificial. This has resulted in a simplified implementation, but also difficulties, such as the partial observability issue due to the lack of complex features in the domain. Most importantly, though, it has limited the possible kinds of goals and drives the agent can have, the kinds of predictions it can makes, and the kinds of appraisals it can generate, which ultimately limits the possible variability in the agent's emotional experience. In the future, we plan to move to a complex, continuous time and space domain. This richer domain should allow us to explore these issues, both with single agents and groups of agents. In short, it will allow us to explore agents that are, ultimately, more emotionally-complex.

Review

In summary, cognitive architecture tells us a lot about emotion. It provides a basis for predicting temporal dynamics, which is essential for making the kind of detailed predictions necessary to validate these theories. It also provides the source of knowledge, including appraisals and coping.

Emotion, on the other hand, gives us insight into cognitive architectures. Existing data highlights possible extensions such as priming effects. New mechanisms are also implied: for example, appraisal theory implies that a cognitive architecture needs an appraisal detector in order to process the generated appraisals. Emotion theories also show a deep interaction between cognition and the body via emotion, mood and feelings. Finally, emotion theories give some insight into the source of drives, situation assessment, conflict detection, coping, and so on.

We have demonstrated a model that shows promise for exploring these aspects of emotion and cognition. We implemented an agent that demonstrated flexible and coherent behavior stemming from different feelings in several environments of varying difficulty. We look forward to extending this architecture in the future.

References

Agre, P. 1988. The dynamic structure of everyday life. Dissertation in Electrical Engineering and Computer Science, MIT.

Damasio, A. (1994). Descartes' error: Emotion, reason, and the human brain. New York: Avon Books.

Damasio, A. (2003*)*. Looking for Spinoza: Joy, sorrow, and the feeling brain. USA: Harcourt.

Forgas, J. P. (1999). Network theories and beyond. In T. Dalgleish & M. Power (Eds.) *Handbook of Cognition and Emotion*. (pp. 591-611). Chichester, England: Wiley & Sons.

Frijda, N. H., Kuipers, P., & ter Schure, E. (1989). Relations among emotion, appraisal, and emotional action readiness. *Journal of Personality and Social Psychology, 57,* 212-228.

Gratch, J., & Marsella, S. (2004). A domain-independent framework for modeling emotion. *Cognitive Systems Research* (5), 269-306.

Gratch & Marsella (in preparation)

Gross, J.J., & John, O.P. (2003). Individual differences in two emotion regulation processes: Implications for affect, relationships, and well-being. *Journal of Personality and Social Psychology*, 85, 348-362.

James, W. (1884). What is an emotion? *Mind*, *9*, 188-205.

Marinier, R., & Laird, J. (2007). Computational Modeling of Mood and Feeling from Emotion. *CogSci 2007.* Nashville: Cognitive Science Society.

Moors 2006

Neal Reilly, W. S. (1996). *Believable Social and Emotional Agents*. (Tech. Rep. CMU-CS-96-138). Pittsburgh, PA: Carnegie Mellon University.

Newell, A. (1990). *Unified Theories of Cognition.* Cambridge: Harvard University Press.

Allen Newell, J. Clifford Shaw, and Herbert A. Simon, Report on a general problem solving program, in Proceedings of the International Conference on Information Processing, UNESCO, Paris, 1960, pages 256-264.

Roseman, I., & Smith, C. A. (2001). Appraisal Theory: Overview, Assumptions, Varieties. In K. Scherer, A. Schorr, & T. Johnstone (Eds.), *Appraisal Processes in Emotion: Theory, Methods, Research.* New York: Oxford Univeristy Press.

Rosenberg, E. L. (1998). Levels of analysis and the organization of affect. *Review of General Psychology*, 2, 247-270.

Scherer, K. (2001). In K. Scherer, A. Schorr, & T. Johnstone (Eds.), *Appraisal Processes in Emotion: Theory, Methods, Research.* New York: Oxford University Press.

Smith, C.A. & Kirby, L.A. (2001). Toward delivering on the promise of appraisal theory. In K.R. Scherer, A. Schorr & T. Johnstone (Eds). *Appraisal Processes in Emotion* (pp. 121-138). Oxford: Oxford University Press.

Smith, C. A. & Lazarus, R. S. (1990). Emotion and adaptation. In L. A. Pervin (Ed.), *Handbook of personality theory and research* (pp. 609-637). New York: Guilford.