BRILL

Multisensory Research 26 (2013) 177–204

MULTISENSORY RESEARCH

brill.com/msr

A Hierarchical Modular Architecture for Embodied Cognition

Dana H. Ballard [1,*], **Dmitry Kit** [1,*], **Constantin A. Rothkopf** [2] **and Brian Sullivan** [3]

[1] Department of Computer Science, University of Texas at Austin, Austin, TX, USA

[2] Frankfurt Institute for Advanced Studies, Goethe University, Frankfurt am Main, Germany

[3] Department of Psychology, University of Texas at Austin, Austin, TX, USA

Received 12 September 2012; accepted 19 October 2012

Abstract

Cognition can appear complex owing to the fact that the brain is capable of an enormous repertoire of behaviors. However, this complexity can be greatly reduced when constraints of time and space are taken into account. The brain is constrained by the body to limit its goal-directed behaviors to just a few independent tasks over the scale of 1–2 min, and can pursue only a very small number of independent agendas. These limitations have been characterized from a number of different vantage points such as attention, working memory and dual task performance. It may be possible that the disparate perspectives of all these methodologies can be unified if behaviors can be seen as modular and hierarchically organized. From this vantage point, cognition can be seen as having a central problem of scheduling behaviors to achieve short term goals. Thus dual-task paradigms can be seen as studying the concurrent management of simultaneous, competing agendas. Attention can be seen as focusing on the decision as to whether to interrupt the current agenda or persevere. Working memory can be seen as the bookkeeping necessary to manage the state of the current active agenda items.

Keywords

Modules, reinforcement, learning, credit assignment, reward

1. Introduction

Very early on it was recognized that to realize the sophisticated decisions that humans routinely make, their brains must utilize some kind of internal model (Tolman, 1948). One of the key figures in the modern day was Neisser, who

* To whom correspondence should be addressed. E-mail: dana@cs.utexas.edu; dkit@cs.utexas.edu

DOI:10.1163/22134808-00002414

championed the idea of an internal cognitive architecture (Neisser, 1967). Subsequent systems codified prior knowledge that came from experts (Anderson, 1983; Laird *et al.*, 1987; Langley and Choi, 2006; Sun, 2006). Such systems use fine-grained rules with variables and bind them by pattern matching. Their broad intent is to cast a behavior as problem solving and find a sequence of rules that will solve it. Despite the enormous difficulties involved, expert systems have achieved notable successes, particularly in intellectual problems where requisite symbol bindings can be readily modeled, such as in algebraic problem solving (Ritter *et al.*, 1998). However, a crucial area where these systems are somewhat dated is that of perception and action. To take a specific example, Anderson's ACT-R appends vision as a module, with the ability to search for parts of the image by coordinates or feature, its rules being based on early ideas from Treisman (1980) and Trick and Pylyshyn (1994) that presuppose that the image can be segmented prior to the introduction of the cognitive agenda.

The regulation of vision and action to secondary levels of importance is consistent with the view of brain function as being composed of separate sequential stages of perception, cognition and action (Arbib, 1998). The characterization has the role of vision to extract information from a scene that is then analyzed by a decision making process that finally orders an appropriate motor response. This view is defended by Ruthruff *et al.* (2003) who point to a cognitive bottleneck in a dual task paradigm. Cognitive limitations block the flow of information from perception to action. However, sequentiality as a strict constraint can be difficult to defend. When following a car in a driving environment, or navigating an obstacle field, the task may be ongoing, with successive actions that have no discrete beginning or end. When responding to cues on a display, decisions are typically done in blocks, so that the expectations as to the response to any particular trial may have been encoded long before its perceptual cues appear. Even in single-trial situations, responses may be driven predominantly by previously-learned priors. Thus the pure perception–cognition–action sequence, while attractive from the point of view of descriptive compactness, may be a rarity in natural behavior.

An alternative to the perception–cognition–action formalism is that of a behavioral primitive, wherein the three components are more integrated. This view is especially associated with Brooks (1986). His original formulation stressed the idea of subsumption, wherein primitive behaviors could be a component of more complicated behaviors by adding additional components that could achieve economies of expression by exploiting, or subsuming, the functions of existing behaviors. This approach has had a huge impact on robotics and cognitive architectures, but has not solved satisfactorily the problem of scaling up to complex situations where many subsuming behaviors start to interact.

To circumvent the scaling dilemma, diverse communities in robotics and psychology have been working on cognitive architectures that take an alternative embodied approach to vision and action and both groups have recognized that the ultimate architecture will have an inhomogeneous hierarchical structure (e.g. Arkin, 1998; Bryson and Stein, 2001; Firby *et al.*, 1995). Robotics researchers in particular have gravitated to a three-tiered structure that models strategic, tactical and detail levels in complex behavior (Bonasso *et al.*, 1997).

One of the most recent foci in cognitive architectures is that of embodied cognition (Adams, 2010; Pfeifer and Scheier, 1999; Shapiro, 2011). Embodied cognition integrates elements from all the robotic advances but in addition places a special stress on the body's role in computation. It emphasizes that the brain cannot be understood in isolation as so much of its structure is dictated by the body it finds itself in and the world that the body has to survive in (Ballard *et al.*, 1997a; Clark, 1999; Noe, 2005; O'Regan and Noe, 2001; Roy and Pentland, 2002; Yu and Ballard, 2004). This viewpoint has important implications for cognitive architectures, because the brain can be dramatically simpler than it could ever be without its encasing milieu. The reason is that the brain does not have to replicate the natural structure of the world or the special ways of interacting with it taken by the body but instead can have an internal structure that implicitly and explicitly anticipates these commitments. Embracing the embodied cognition paradigm is not without risks. The very abstract levels of description (Stewart *et al.*, 2010; Vareala *et al.*, 1991) risk losing a direct connection to the human body's particulars. On the other hand, starting with very low level models, e.g. at the neural level (Nordfang *et al.*, 2012), faces difficulties in managing the brain's enormous numbers of degrees of freedom. We attempt to choose a middle ground of abstract description that explicitly addresses issues that of scheduling information acquisition using a foveal retina, but our models will ultimately prove inadequate if they cannot eventually be further grounded in the body's neural control systems.

This paper describes a specific hierarchical cognitive architecture that uses small collections of special-purpose behaviors to achieve short-term cognitive goals. These behaviors are defined at the intermediate tactical level of cognitive representation in terms of modules. Modules are self-contained and can operate independently but, most importantly, they can be composed in small groups. While the specification of modules could potentially take several forms, we use reinforcement learning (RL) as a basis for defining them here. The RL formalism has several advantages, two of which will be highlighted here. Firstly, the formalism provides an understanding for task directed fixations as a method of increasing reward by reducing task uncertainty. Secondly the formalism provides a way of computing the value of new tasks by executing them within the context of subsets of other tasks whose reward values have been already estimated. In addition the RL formalism provides a way

of estimating rewards from observations of subject data. Technical portions of this paper appeared in Rothkopf and Ballard (2009, 2010) and in Rothkopf *et al.* (2007). The primary thrust of the paper is to relate these technical features to psychological notions of attention, working memory and dual task performance.

2. A Hierarchical Cognitive Architecture

Newell pointed out that any complex system that we know about is organized hierarchically. Furthermore as one proceeds up the hierarchy towards more abstract representations, its combined components take up more space, and run more slowly, simply because the signals have further to travel. In silicon we have the ready hierarchy of: gates, VLSI circuitry, microcode, assembly language, programming language, operating system. For each level, a crucial understanding is that one must respect the abstractions at that level. For example in microcode, multiplication is deconstructed into sifts and adds and those parts can be inspected. At the level of assembly language, this can no longer be done as multiplication is a primitive.

Although it is possible to define several more abstract levels (Hurley, 2008), our hierarchy consists of four levels — Debug, Operating System, Task and Routines — but in this paper we focus on the Task level. When something goes very wrong, the job of the Debug level attempts to find out what is wrong and reprogram the responsible modules. This would be loosely characterized as high-level attention. The Operating System level has the job of selecting an appropriate suite of modules, including an appropriate alerting module, for the current set of behavioral goals. A Task is a module dedicated to achieving one of these goals. The task module may have a succession of states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$, each of which can be interrogated by sensory routines and directed by motor routines. The routines level comprises sensory and motor routines that respectively compute states and execute actions. Each action make earn a scalar reward $r \in \mathcal{R}$ that may be positive or negative. Figure 1 summarizes this organization.

To appreciate the value of our particular hierarchical organization, it helps to review some key concepts related to attention as conceptualized in psychology. In particular, the triage of attention used by (Fan *et al.*, 2005; Posner and Rothbart, 2007) is particularly helpful. They characterize attention as having three separate neural networks that handle alerting, orienting, and executive functions. These functions have also been associated with specific neurotransmitters norepinephrine, acetocholine and dopamine respectively. To summarize the psychological view: Alerting is associated with sensitivity to incoming stimuli; orienting is the selection of information from sensory input, and executive attention involves mechanisms for resolving conflict. It turns out

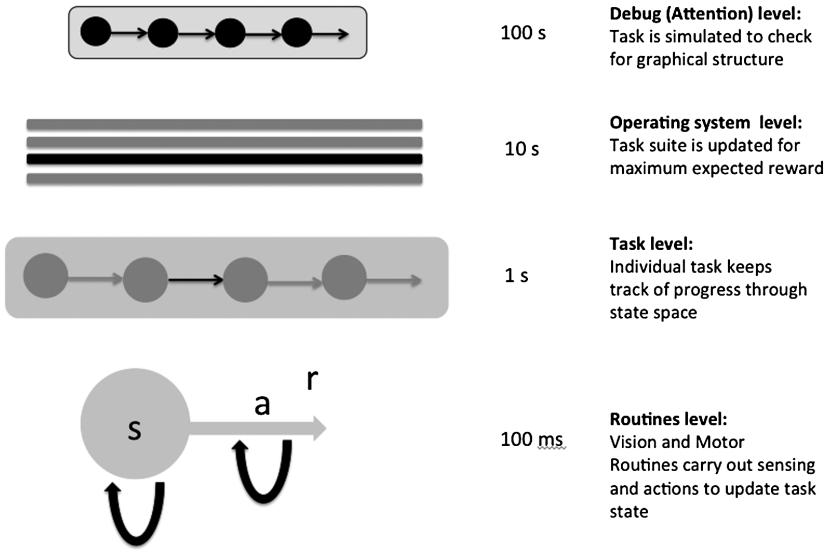

Figure 1. Four levels of a hierarchical cognitive architecture that operate at different timescales. The central element is the task level, wherein a given task may be described in terms of a module of states and actions. A *thread* keeps track of the process of execution though the module. The next level down consists of visual and motor routines (arrows) that monitor the status state and action space fidelities respectively. Above the task level is the operating system level, whereby priorities for modules are use to select an appropriate small suite of modules to manage a given real world situation. The topmost level is characterized as an attentional level. If a given module is off the page of its expectations, it may be re-programmed via simulation and modification.

that each of these components of the broad characterization of attention can be seen as addressing an implementation issue in the modules cognitive architecture. However, at the specific tactical abstraction level of modules, the specific enfleshment of these concepts results in slightly different interpretations from those used in the psychological domain.

Our central hypothesis is that any ongoing behavior can be seen as a possibly dynamic composition of a small number of active modules, each of which is associated with a separate task. For example in driving a car in a city, different modules could be active for staying in a lane, monitoring for pedestrians, and dealing with other traffic. A central constraint is that of capacity. It is assumed from neural principles that the number of instantaneously co-active modules must be kept within a small bound. Thus we would reinterpret Baddeley's classical result on the limitations of working memory (Baddeley, 1992) to be about that state information necessary to keep track of different co-active modules. Computer science uses the term *thread* to denote the unique state needed to describe a process, and in that sense working memory is about threads.

Following Luck and Vogel (1997), we take the limit on the number of coactive modules to be about four. The exacting constraint of a small number of modules places a premium on a method for choosing between them, and thus provides the primary raison d'être for an executive. The executive's function is to evaluate the effectiveness of the existing modules with respect to other modules waiting in the cognitive wings to see if there are grounds for a productive exchange.

In order for the executive to evaluate the prospects for an inactive module, there must be a calculation of these prospects that falls short of activating the module. There are many possible ways to do this. Our module design posits a two-fold index that (1) expresses that relevance of a module, and (2) estimates the worth of its potential contribution. These calculations would be putatively handled by a special alerting module whose function would be to trap any of a set of unexpected conditions. In this description, we are close to a saliency map (Itti, 2005; Itti and Koch, 2000), but our saliency map needs to have the combination of features, priors (Torralba *et al.*, 2006) and task relevance measures. The focus of psychology has been on exogenous visual features on the optic array, but natural behaviors are mediated by many factors. The signaling feature may be in the form of an auditory, tactile or olfactory cue and its value may be signaled by a variety of factors that may have their roots in endogenous as well as exogenous sources. In our modular scheme, a special alerting module provides a short list of contenders and the executive decides which ones are in or out.

The final issue is associated with the set of active modules. The virtue of a module is that it is a program fragment that encapsulates detailed knowledge about dealing with the world to achieve a specific goal. In this effort, it usually deploys very specialized resources for extracting just the knowledge it needs from the surround (Roelfsema *et al.*, 2003; Ullman, 1985). To take a visual example, a visual routine, deployed by a module, may need the color of a region of space and in that case, may need to deploy gaze to obtain a measurement with enhanced foveal resolution. If several active modules require the use of the gaze vector, there needs to be an orienting mechanism, both to resolve this competition and to obtain the desired measurement.

3. Evidence for Routines

Although we know enough to immediately reject the idea of an image in the head, the parcellation of visual features into the levels of the cortical hierarchy may tempt us to posit some internal feature based Cartesian Theater. Instead a functional hypothesis is that the information stored in the cortex is indeed some kind of model for the world, but that the model is continually being simulated, calibrated and updated with new measurements. Much evi-

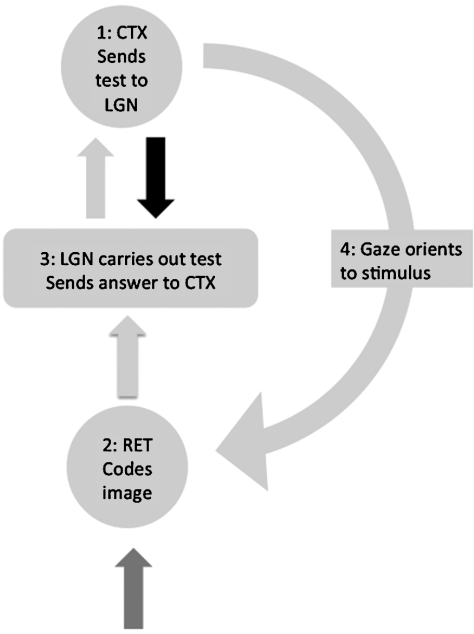

Figure 2. In a task-directed module, the maintenance of state information is handled by routines that exhibit agenda-driven control strategies. To get information, a hypothesis to be tested is putatively sent to the thalamus, where it is compared to coded image data. The result is state information that may in addition trigger a gaze change.

dence suggests that these measurements are carried out with fixations, with specific quantile pieces of information acquired at each given fixation. Figure 2 shows our basic architecture, which has a given test 'downloaded' to the Lateral Geniculate Nucleus and applied to coded image data coming from the retina. The important point is that the test is typically sent 'down' before the fixation is established for reasons of speed efficiency.

Direct measurements of brain activity provide a plethora of evidence that the segments in a task take the form of specialized modules. For example, the Basal Ganglia circuitry shows specific neural circuits that respond to short components of a larger task (Hikosaka *et al.*, 2008; Shultz *et al.*, 1997b).

Moreover, embodied cognition studies provide much additional evidence. One compelling example is that of Rothkopf and Ballard (2009) that measures human gaze fixations during the navigation task that has separate trophic (picking up litter objects) and anti-trophic (avoiding obstacles) components. The overall setting is that of our virtual environment and uses identical litter and obstacle shapes that are only distinguished by object color. When picking up the object as litter, subjects' gaze is allocated to the center of the object, but when avoiding the same object, subjects' gaze fall on the objects' edges.

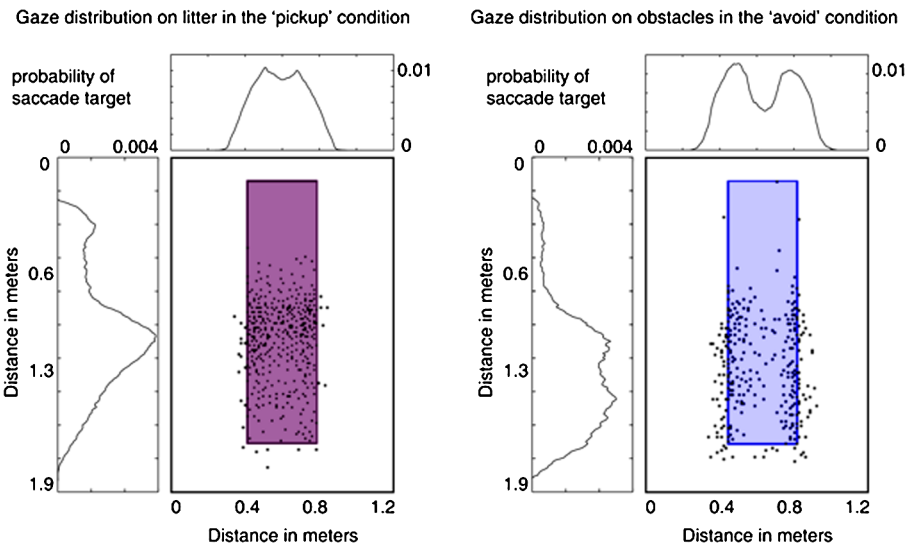

Figure 3. Human gaze data for the same environment showing striking evidence for visual routines. Humans in a virtual walking environment manipulate gaze location depending on the specific task goal. The small black dots show the location of all fixation points on litter and obstacles. When picking up litter (left) gaze points cluster on the center of the object. When avoiding a similar object (right) gaze points cluster at the edges. From Rothkopf and Ballard (2009). This figure is published in color in the online version.

Figure 3 shows both of these cases. The inference is that when approaching an object, subjects use the expanding optic flow field to home in on it, but when avoiding an object, subjects use the different strategy of rotating about an edge of it. These two different strategies would be efficiently handled by individual task solutions, i.e. different modules, so that individual solutions can be learned and reused in combinations.

The example shown in Fig. 3 just points out a standard way of purposefully interacting with the environment, but there are many others. Subjects manipulating blocks in another virtual world code the color of a block in a single fixation at the beginning of a trial and then are apt to miss catch trial color changes to the block, even when cued to look for changes (Droll *et al.*, 2005). Subjects searching for an object in a large image using a remembered preview of the image have very different scanning patterns than subjects who are just given a small image icon containing the target (Rao *et al.*, 2002). Subjects filling a cup with a soda lock their gaze on the rising liquid level during the operation to use the level as a termination condition (Hayhoe *et al.*, 2003).

The routines approach to vision also provides a ready explanation of the 'attentional blink' phenomenon, which is much simpler than current theories (e.g. Bowman and Wyble, 2007). In visual attentional blink, the second

of two successive visual stimuli is missed if it follows the first by approximately 300 milliseconds. In terms of an active visual system running routines, 300 milliseconds corresponds to the modal fixation time, so the blocking of the second stimulus is a Nyquist phenomenon. The rate of stimuli to be tested is too close to the natural sampling rate used by the routines (see Note 1).

4. Modules and the Executive

Evidence for the suitability of the modules model comes from both simulations and experiment. Sprague *et al.* (2007) found that subjects walking down a sidewalk in a virtual environment focused their fixations almost exclusively on the task related objects. In a follow-on experiment, Rothkopf and Ballard showed that when presented with distractors, subjects would fixate them initially, but once they were told to start walking down the sidewalk and carry out the other sub-tasks of collecting litter and avoiding obstacles, they ignored the distractors almost completely, as shown in Fig. 4. Evidence such as this motivates the use of modules. The particular setting we use, also motivated by biological factors is that of reinforcement learning (RL).

The central constraint for our RL setting is that the individual rewards due to each module are not known, but only the global reward is supplied to the agent at each time step. Using only this information, the agent needs to compute the share of the credit for each module. To describe this situation formally, we can modularize the definition of a Markov Decision Process (MDP) given in Appendix A to:

$$M_i = \{\mathcal{S}_i, \mathcal{A}, T_i, G_{\mathcal{M}}\} \tag{1}$$

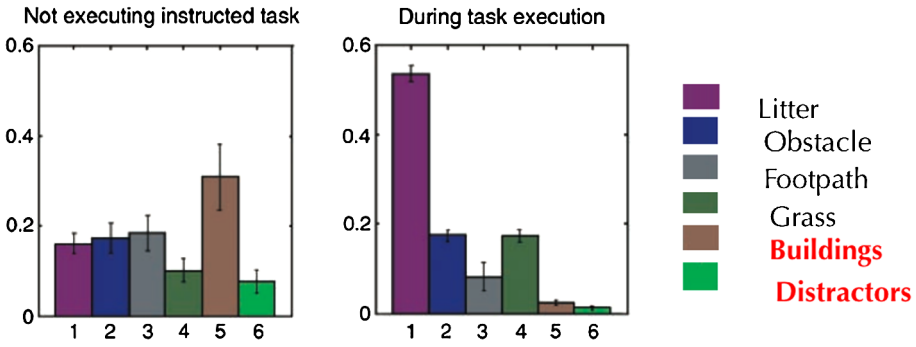

Figure 4. In walking down a sidewalk, the virtual reality scenery was augmented with a dizzying array of distractors e.g. an upside-down cow. Subjects viewed the scene with a binocular HMD which was upadate for head motion. While waiting for a start command, subjects did fixate distractors (items 5 and 6), but when the sidewalk navigation experiment began, subjects fixated the task-relevant objects almost exclusively (items 1, 2 and 3).

where the subscript $\mathcal{M}$ denotes that at each time step, there may be received reward G that is a function of the modules that are active. It is important to note that the set of states of a module, $\mathcal{S}$, are just the information it needs to specify the requisite set of actions $\mathcal{A}$. Thus they are in the spirit of Gibson's affordances (Gibson, 1979), and may not refer to individual objects or features. Also it is important to note that, in our embodiment model, the action selected is shared across the modules, e.g. one can only walk in one direction at a time.

Our central assumption is that an overall complex problem can be factored into a small set of MDPs, but any given factorization can only be expected to be valid for some transient period. Thus, the set of active modules is expected to change over time as the actions taken direct the agent to different parts of the composite state space. This raises two issues that we finesse: (1) How is a module activated? We assume that the sensory information provides a trigger as to when a module will be helpful. (2) How many modules can be active at a time? Extensive research on the capacity of humans to multi-task suggest that this number might be small, approximately four (Luck and Vogel, 1997). Taking both these constraints into consideration in our simulations, we use trigger features and use the value of four as a bound on the number of simultaneously active modules. Although this module activation protocol will allow the modules to learn as long as they sample their state–action spaces sufficiently often, there is still the question of how often to use it. If it is used at every time step, the modules chosen will have little time to explore their problem spaces and adjust their Q values. At the same time, if a set of modules is invariant for too long, the agent may not be able to adjust to important environmental exigencies. Thus for the length of module activation, we introduce the constraint of an episode of fixed length parameter Δ (see Fig. 5). During each episode, only a subset of the total module set is active. The guiding hypothesis is that in the time-course of behavior, a certain set of goals is pursued and therefore the corresponding modules that are needed to achieve these goals become active and those that correspond to tasks that are not pursued become inactive (Sprague *et al.*, 2007). During an episode, the composition of a particular module set is assumed to not change. Given this constraint, the pivotal idea is that, within each episode, each active module can refine its own reward estimates by having access to the sum of the reward estimates of the other active modules.

In summary, the properties of the executive are as follows:

1. The overall behavior of modules is such that they work together in different subsets for a set time Δ so that the reward estimates can be learned;
2. The sum of the current estimates of the reward across an entire subset is accessible to each individual module in the subset at each moment by assumption;

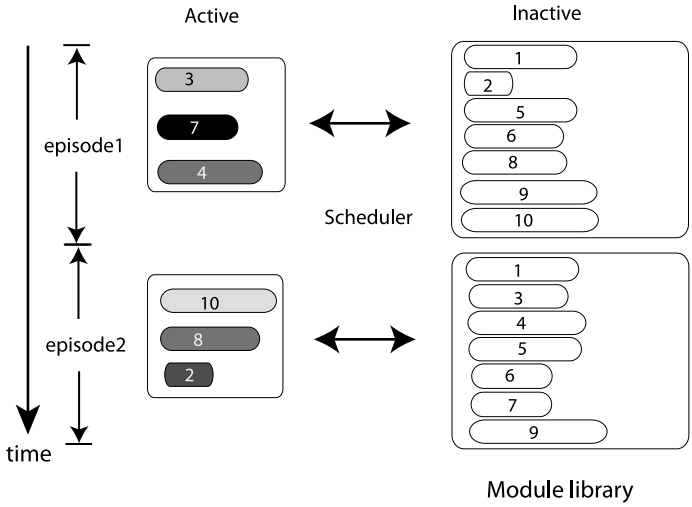

Figure 5. In any period during behavior there is only a subset of the total module set that is active. We term these periods *episodes*. In the time course of behavior, modules that are needed become active and those that are no longer needed become inactive. The diagram depicts two sequential episodes of three modules each {3, 4, 7} and {2, 8, 10}. The different modules are denoted with different shadings and numbers. The different lengths indicate that modules can exhibit different numbers of states and finish at different times. The horizontal arrows denote the scheduler's action in activating and deactivating modules. On the right is the large library of possible modules. Our formal results only depend on each module being chosen sufficiently often and not on the details of the selection strategy. The same module may be selected in sequential episodes.

3. The sampled subsets collectively must span the module space because the reward calculations demand this. The consequences of a module being activated are that:

4. It has used an associated procedure, such as a visual routine (Ballard *et al.*, 1997b; Ullman, 1985), to compute the initial state the module is in. In our examples we assume or supply a routine that does this;

5. Its Q-values are included in the sum indicated in equation (17) used to select an action, and

6. It influences the global reward that is handed out at every time step.

The specification of the executive is the least satisfactory component of the cognitive architecture of several reasons. Foremost is that there is very little data of time and motion studies that includes eye, head and hand movements with sufficient fidelity so that we could compare it to our model. In our simulations additional effort needs to be done to shape the transient behavior of the executive in the case where there would be a library of modules of inter-

esting proportions. Another issue is that the fixed parameter Δ is *ad hoc*, and has been chosen mostly to provide a formal scaffold for the credit assignment study in Section 7. Also we need to integrate and test an agenda-switching module (see the following section). Thus all in all the executive description has merit in providing a structure for attaching the modules, but many of its features need further development.

5. Alerting

As an example of one role of the executive, consider the problem of dealing with exigencies during freeway driving, an example of which is shown in Fig. 6. Freeway driving is characterized by a very large motion stimulus, but for the most part that stimulus's main component is the radial motion produced by the parallel trajectories of the driver's and surrounding vehicles. Nonetheless there are interrupts in this pattern that must be dealt with. In the case of a car in front encroaching on the driver's lane, the event must be detected by an alerting system and then the executive must switch the requisite module into the active module set.

Although this has only been tested in the case of driving (see Fig. 6), our model of attention would use a special module whose function would be to test the environment for situations like this one that require a change of agenda. These changes span two levels of the cognitive hierarchy. If the change easily can be handled by another module, the suite of modules is updated. However, if not, it must be handled at the level of conscious awareness, which can resort to simulations to diagnose a more elaborate response.

6. Modules and Gaze Arbitration

Independent modules provide another boon in the embodied cognition setting, as they provide an elegant motivation for the disposition of gaze. Owing to the small visual angle of the human fovea, approximately one degree, gaze is not easily shared in servicing different tasks, and must be allocated amongst them. Arbitrating gaze requires a different approach than arbitrating control of the body. Reinforcement learning algorithms are best suited to handling actions that have direct consequences for a task. Actions such as eye fixations are difficult to put in this framework because they have only indirect consequences: they do not change the physical state of the agent or the environment; they serve only to obtain information.

As Sprague *et al.* (2007) show, a much better strategy than the straightforward RL protocol is to choose to use gaze to service the behavior that has the most to gain by being updated. The advantage of doing so is that uncertainty in the state information is reduced, leading to better policy choices. Absent

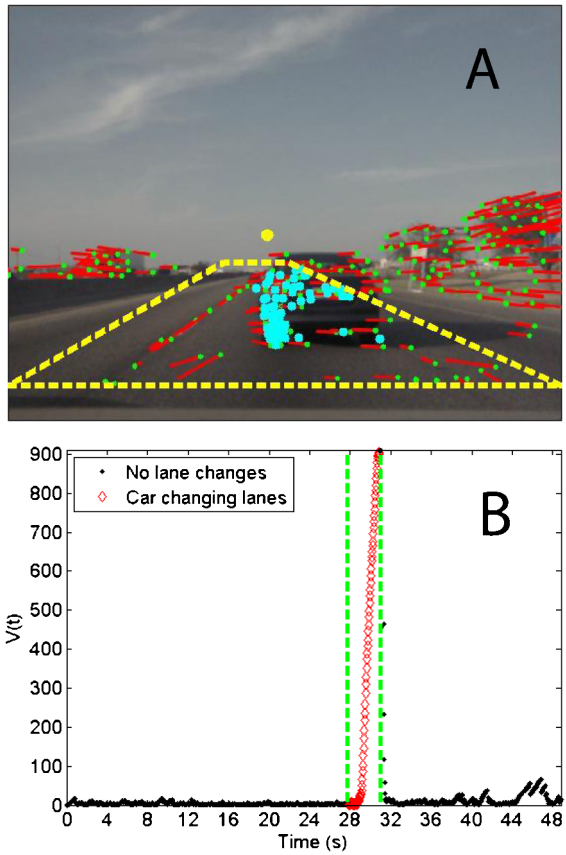

Figure 6. A potential job for an alerting module: Detecting unusual variations in optic flow while driving. (A) Encroaching car produces a pronounced deviation from background radial flow expectation. Radial flow can be dismissed as a normal expectation, but the horizontal flow of a car changing lanes signals an alert. (B) The time line shows that this signal, as measured by a space and time-window integration, is easily detectable. This figure is published in color in the online version.

these updates, as time evolves, the uncertainty of the state of a behavior grows owing to noise in the human environment dynamics, introducing the possibility of low rewards. Deploying gaze to measure the state in a particular module more accurately reduces this risk for that module.

Estimating the cost of uncertainty is equivalent to estimating the expected cost of incorrect action choices that result from uncertainty. Assuming that the expected rewards for an action selection (coded in Q functions) are known, and that Kalman filters can provide the necessary distributions over the state variables, it is straightforward to estimate this factor, $loss_b$, for each behavior 'b' by sampling, using the following analysis. The loss value can be broken

down into the losses associated with the uncertainty for each particular behavior b:

$$loss_b = E\left[\max_a\left(Q_b(s_b, a) + \sum_{i\in B, i\neq b} Q_i^E(s_i, a)\right)\right] - \sum_i Q_i^E(s_i, a_E). \quad (2)$$

Here, the expectation on the left is computed only over s_b. The value on the left is the expected return if s_b were known but the other state variables were not. The value on the right is the expected return if none of the state variables are known. The difference is interpreted as the cost of the uncertainty associated with s_b. The maximum of these values is then used to select which behavior should be given control of gaze.

To summarize this use of gaze: Besides the executive and alerting facets of attention, the third important aspect is that of orienting. A module's resources must be focused to acquire the information necessary for its function. The set of active rules formulation used here places a burden on the orienting task of any given visual module as the resource often necessary for its function may be competed for across the current module set and this is the level tackled by the Sprague model. Figure 7 shows that that resolving this competition by allocating gaze that would reduce its reward-weighted uncertainty the most is a superior strategy to standard methods of gaze allocation.

7. Online Module Calibration

The modular RL-based paradigm has the benefit of allowing a complex behavior to be broken down into small manageable parts, but leaves open a large potential problem of assigning reward values to each of the parts. How should the appropriate reward value for a module be assigned? Fortunately, this problem can be handled algorithmically as long as the behaving system has access to the total reward of the suite of behaviors at any moment.

Each active module represents some portion of the entire state space and executes some part of the composite action, but without some additional constraint they only have access to a global performance measure, defined as the sum of the individual rewards collected by all of the $\mathcal{M}$ active modules at each time step:

$$G_t = \sum_{i\in\mathcal{M}} r_t^{(i)}. \quad (3)$$

The central problem that we tackle is how to learn the composite Q values $Q^{(i)}(s^{(i)}, a)$ when only global rewards G_t are directly observed, but not the individual values $\{r_t^i\}$ (see Fig. 8).

The key additional constraint that we introduce is an assumption that the system can use the sum of rewards from the modules that are co-active at

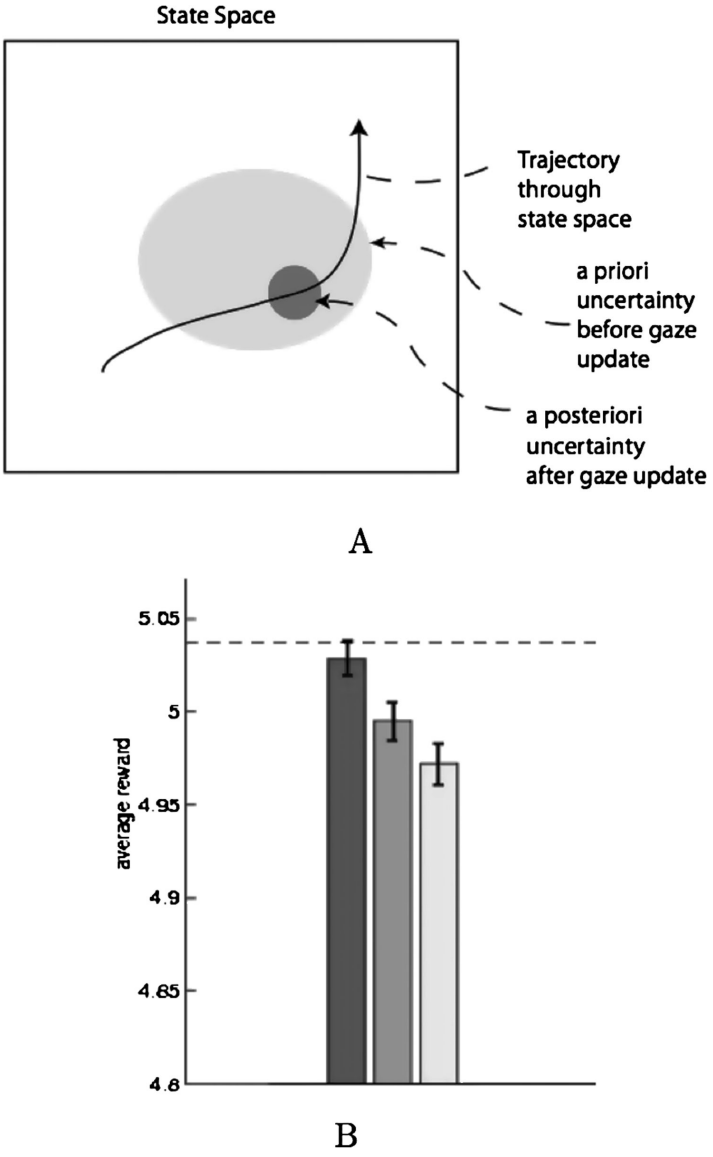

Figure 7. (A) The Sprague model of gaze allocation. Modules compete for gaze in order to update their measurements. The figure shows a caricature of the basic method for a given module. The trajectory through the agent's state space is estimated using Kalman filter that propagates estimates in the absence of measurements and, as a consequence, build up uncertainty (large shaded area). If the behavior succeeds in obtaining a fixation, state space uncertainty is reduced (dark). The reinforcement learning model allows the value of reducing uncertainty to be calculated. (B) In the side-walking venue, three modules are updated using the Sprague protocol, a sequential protocol and a random protocol (reading from left to right). The Sprague protocol outperforms the other two.

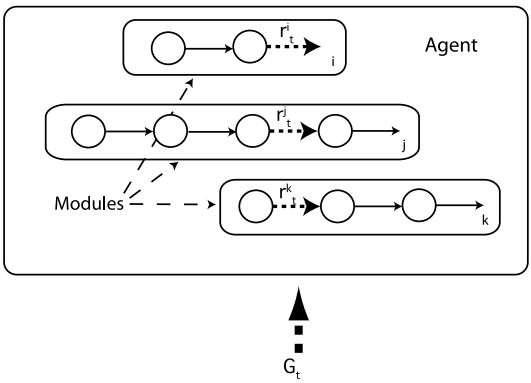

Figure 8. A fundamental problem for a biological agent using a modular architecture. At any given instant, shown with dotted lines, when multiple modules are active and only a global reward signal G is available, the modules each have to be able to calculate how much of the rewards is due to their activation. This is known as the credit assignment problem. Our setting simplifies the problem by assuming that individual reinforcement learning modules are independent and communicate only their estimates of their reward values. The modules can be activated and deactivated asynchronously, and may each need different numbers of steps to complete, as suggested by the diagram.

any instant. This knowledge leads to the idea to use the different sets to estimate the difference between the total observed reward G_t and the sum of the current estimates of the individual rewards of the concurrently running behaviors. Credit assignment is achieved by bootstrapping these estimates over multiple task combinations, during which different subsets of behaviors are active. Dropping the temporal subscript for convenience, this reasoning can be formalized as requiring the individual behaviors to learn independent reward models $r^{(i)}(s^{(i)}, a)$. The current reward estimate for one particular behavior i, is obtained as

$$\hat{r}^{(i)} \leftarrow \hat{r}^{(i)} + \beta \delta_{r^{(i)}}, \tag{4}$$

where the error on the reward estimates δ_r is calculated as the difference between the actual global reward received and the estimated global reward calculated from the sum of the component estimates which can be informatively rearranged as in equation (5):

$$\hat{r}^{(i)} \leftarrow (1-\beta)\hat{r}^{(i)} + \beta\left(G - \sum_{j \in \mathcal{M}, j \neq i} \hat{r}^{(j)}\right). \tag{5}$$

To interpret this equation: each module should adjust its reward estimate by a weighted sum of its own reward estimate and the estimate of its reward inferred from that of the other active modules. Together with the module activation protocol and Δ, equation (5) represents the core of our solution to the

credit assignment problem. When one particular subset of tasks is pursued, each active behavior adjusts the current reward estimates $\hat{r}_i$ in the individual reward functions according to equation (5) at each time step. Over time, the set of tasks that have to be solved will change, resulting in a different set of behaviors being active, so that a new adjustment is applied to the reward functions according to equation (5). This bootstrapping process therefore relies on the assertion that the subsets of active behaviors visit all component behaviors.

The component Q values for the state–action pairs of the individual behaviors are learned using the above estimates of the individual reward functions. Given the current reward estimates obtained through repeated application of equation (5), the SARSA algorithm is used to learn the component Q-functions:

$$Q_i\left(s_t^{(i)}, a_t^{(i)}\right) \leftarrow Q_i\left(s_t^{(i)}, a_t^{(i)}\right) + \alpha\delta_{Q_i}, \tag{6}$$

where δ_{Q_i} now contains these estimates $\hat{r}_t^{(i)}$ and is given by equation (7):

$$\delta_{Q_i} = \hat{r}_t^{(i)} + \gamma\, Q_i\left(s_{t+1}^{(i)}, a_{t+1}^{(i)}\right) - Q_i\left(s_t^{(i)}, a_t^{(i)}\right). \tag{7}$$

The usage of an on-policy learning rule such as SARSA is necessarily an approximation as noted in (Sprague and Ballard, 2003), because the arbitration process specified by equation (17) may select actions that are suboptimal for one or more of the modules. A feature of the SARSA algorithm is that it makes use of suboptimal policy decisions.

A concern one might have at this point is that since the rewards and the policies based on them are varying in separate algorithms, the net result might be that neither estimate converges. However it can be proved that this is not the case and that furthermore convergence in the reward space is very rapid, as shown by the following example.

Calibrating reward in walkway navigation problem we demonstrate the algorithm on the problem described earlier of an agent in a simulated three-dimensional world walking on a sidewalk while avoiding obstacles and picking up litter (Sprague and Ballard, 2003). However, that solution was obtained by delivering each of the individual learners their respective reward; that is, the agent received three separate rewards, one for the walkway following module, one for the obstacle avoidance module, and one for the litter picking up module (Fig. 9). This problem was re-coded here but with the additional constraint of only global reward being observed by all modules in each task combination. The global reward was always the linear sum of the rewards obtained by the individual modules according to equation (3).

For all these simulations, the RL learning parameter α was 0.1. The first experiment uses both constant β values from the set $\{0.01, 0.1, 0.2, 0.5\}$ as well as the variance weighted of the formula for calculating β (Rothkopf and Ballard, 2010). The experiment involving learning to navigate along a path while

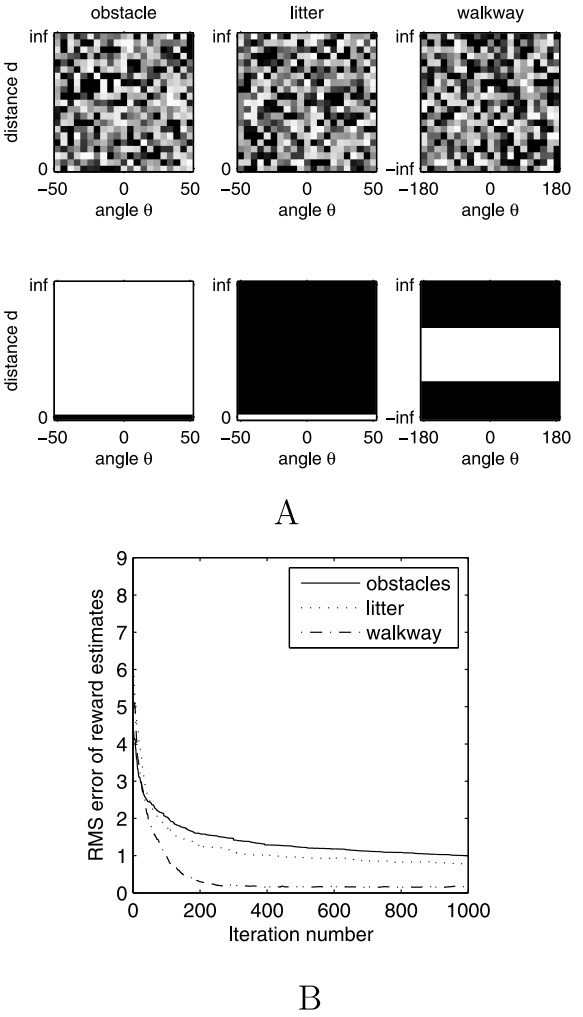

Figure 9. (A) Reward calculations for the walkway navigation task for the three component behaviors. Top row: Initial values. Bottom row: Final reward estimates. (B) Time course of learning reward for each of the three component behaviors. RMS error between true and calculated reward as a function of iteration number.

avoiding obstacles and approaching targets uses a constant β value of 0.1. The distance is scaled logarithmically similarly to the original setup (Sprague *et al.*, 2007) and the resulting distance d_i is then discretized into 21 possible values between 0 and infinite distance. The angles within the field of view, i.e. with a magnitude smaller than 50 degrees are similarly discretized to 21 values. The walkway state space is slightly different from (Sprague *et al.*, 2007) in that it represents all positions of the agent relative to the walkway for all possible

walking directions. Finally, instead of 3 possible actions as in Sprague *et al.* (2007), the current simulations use 5 actions corresponding to steering at one of the five angles $\{-15, -7.5, 0, 7.5, 15\}$ with additive Gaussian noise of variance $\sigma = 1$. To learn policies and Q values, different subsets of modules were selected for different episodes and the correct global reward supplied for each individual subset.

The basic time unit of computation was chosen to be 300 ms, which is the average duration of a fixational eye movement. Human subjects took an average duration of 1 min 48 s to carry out these tasks, which is approximately 325 intervals of 300 ms. Therefore, an individual episode consists of 325 discrete time steps. At the beginning of each episode it is determined which tasks have high priority. During each episode, it is equally probable that either two or three tasks are pursued. The reward values displayed as a function of the state space locations are shown in Fig. 7A. Starting from random values and receiving only global reward at each step, the agent's modules are able to arrive at good estimates of the true reward. The accuracy of these estimates is shown in Fig. 7B.

8. Discussion and Conclusions

This paper summarizes earlier work on an embodied cognition model based on behavioral modules. In the module formalism, individual task solutions can be learned by specialized modules with independent state variables. The focus of this paper is to relate this methodology to ongoing work in experimental psychology as it turns out that there are several convergent concepts between the two research streams. In particular the psychological attention triage of executive, alerting and orientation has direct parallels in the modules formalism's concepts of executive, indexing, and routines, respectively. And the notion of working memory can be loosely corresponded to the different states needed to individuate different modules.

Echoing Newell, we assert that the brain must have a computational hierarchy in order to organize the complexity of its computations. The ultimate model will probably be different from the one advocated herein, but whatever the final model, it must address the same questions. Furthermore the levels here suggest simpler viewpoints than the ones prevalent in more psychologically based paradigms. For example the use of 'items' for the contents of working memory instead of 'threads' gets into trouble when items are 'chunked', a vague ill-defined term. And the explanation of the attentional blink as a consequence of multi-tasking temporal sampling constraints is a least more straightforward than more involved models that require elaborate models of working memory. Like hierarchical organizations in silicon, hierarchical mod-

els of the brain's computation can potentially resolve technical problems at different levels of abstraction, resulting in simpler overall descriptions.

The modules formulation relies on the agent carrying out multiple task combinations over time, which enables the correct learning of individual rewards for the component tasks. Accordingly, carrying out multiple concurrent task combinations is not a complication but enables learning about the rewards associated with individual tasks. The key constraints, motivated by the need for a system that would potentially scale to a large library of behaviors, are (1) the overall system must be structured such that the system could achieve its goals by using only a subset of its behavioral repertoire at any instant, (2) the reward gained by this subset is the total of that earned by its component behaviors, and (3) the modules must be used in linearly independent combinations. The use of modules allows the rewards obtained by reinforcement to be estimated on-line. In addition this formulation lends itself to use the uncertainties in current reward estimates for combining them amongst modules, which speeds convergence of the estimating process.

The linear independence constraint is important as, without it, the Q values and their corresponding value functions V cannot be correctly computed. Thus, although it may be possible to learn some of the policies for component modules for one particular task combination without it, the value functions may be corrupted by a large bias, which will be especially problematic when new task combinations are to be solved. The reward estimates will be biased such that they have to be relearned, but will again be biased.

In our venue, small numbers of behaviors that are appropriate for the current situation are selected in an on-line fashion. In this situation it is essential to get the Q-values right. An algorithm that models other modules' contribution as pure noise will compute the correct policy when all the behaviors/agents are active but this result will not extend to active subsets of modules and behaviors because incorrect Q values, when used in subsets, will cause chaotic behavior.

The modules formulation is related to earlier approaches that start out with compositional solutions to individual problems and then devise methods in order to combine a large number of such elemental solutions e.g. Singh and Cohn (1998) and Meuleau *et al.* (1998). Both approaches are concerned with learning solutions to large MDPs by utilizing solutions or partial solutions to smaller component MDPs. In (Meuleau *et al.*, 1998) the solutions to such components are heuristically combined to find an approximate solution to the composite MDP by exploiting assumptions on the structure of the joint action space. A way of learning a composite MDP from individual component MDPs by merging has been described in Singh and Cohn (1998). However, the composite problem is solved in a more *ad hoc* way using bounds on the state values derived from the state values of the individual component MDPs.

Attempts to overcome the scaling problem in more elegant ways than *ab initio* factoring try to exploit inherent structure in the problem (Barto and Mahadevan, 2003; Dayan and Hinton, 1992; Parr and Russell, 1997; Sutton *et al.*, 1999; Vigorito and Barto, 2010). Factoring can use graphical models that express conditional independencies can reduce the size of the variables necessary for a full description of the problem at hand (Craig *et al.*, 2000; Guestrin *et al.*, 2003). The approach by Sallans and Hinton (2004) can also be conceptualized as exploiting the statistical structures of the state and action spaces. Doya *et al.* (2002) and Samejima *et al.* (2003) use a number of actor-critic modules and learn a linear combination of the controllers for the local approximation of the policy. All these methods constitute advances and our method is extensible to them to the extent that they can be encapsulated into modules that are made explicit and the issues related to module activation are addressed.

In summary, and returning to the larger issues of the paper, the overall goals of the paper have been twofold. Firstly we have advocated that in order be accessible a comprehensive model of human behavior during common tasks should have hierarchical levels, with different functions tackled at different levels of the hierarchy. The specific hierarchy proposed here is best seen as provisional, but has served the purposes of making connections to basic and important concepts in psychological research, as well as providing a scaffold for the centerpiece of MDP-based modules. Composing these modules provide a flexible way of accounting for the kinds of coordination structure observed in everyday human physical behavior that can be tested experimentally.

Acknowledgements

The research reported herein was supported by NIH Grants RR009283 and MH060624, as well as NSF grant 0932277.

Notes

1. Explanation courtesy of Wolfgang Einhuser-Treyer.
2. SARSA is an acronym for the quintuple $s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}$ denoting the actual trajectory followed.

References

Adams, F. (2010). Embodied cognition, *Phenomenol. Cogn.* **9**, 619–628.

Anderson, J. (1983). *The Architecture of Cognition*. Harvard University Press, New Haven, CT, USA.

Arbib, M. A. (1988). *The Handbook of Brain Theory and Neural Networks*, pp. 830–834. MIT Press, Cambridge, MA, USA.

Arkin, R. (1998). *Behavior Based Robotics*. MIT Press, Cambridge, MA, USA.

Baddeley, A. (1992). Working memory, *Science* **255**, 556–559.

Ballard, D. H., Hayhoe, M. M., Pook, P. and Rao, R. (1997a). Deictic codes for the embodiment of cognition, *Behav. Brain Sci.* **20**, 723–767.

Ballard, D. H., Hayhoe, M. M., Pook, P. K. and Rao, R. P. N. (1997b). Deictic codes for the embodiment of cognition, *Behav. Brain Sci.* **20**, 723–742.

Barto, A. G. and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning, *Discrete Event Dyn. S.* **13**, 41–77.

Bonasso, E. P., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. P. and Slack, M. G. (1997). Experiences with an architecture for intelligent reactive agents, *J. Exp. Theor. Artif. Intell.* **9**, 237–256.

Bowman, H. and Wyble, B. (2007). The simultaneous type, serial token model of temporal attention and working memory, *Psychol. Rev.* **114**, 38–70.

Brooks, R. (1986). A robust layered control system for a mobile robot, robotics and automation, *IEEE J.* **2**, 14–23.

Bryson, J. J. and Stein, L. A. (2001). Modularity and design in reactive intelligence, in: *Internat. Joint Conf. on Artificial Intelligence*, Seattle, Washington, USA.

Clark, A. (1999). An embodied model of cognitive science? *Trends Cogn. Sci.* **3**, 345–351.

Craig Boutilier, R. D. and Goldszmidt, M. (2000). Stochastic dynamic programming with factored representations, *Artifi. Intell.* **121**, 49–107.

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B. and Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans, *Nature* **44**, 876–879.

Dayan, P. and Hinton, G. E. (1992). Feudal reinforcement learning, *Neural Information Processing Systems* **5**, 271.

Doya, K., Samejima, K., Katagiri, K. I. and Kawato, M. (2002). Multiple model-based reinforcement learning, *Neural Comput.* **14**, 1347–1369.

Droll, J., Hayhoe, M., Triesch, J. and Sullivan, B. (2005). Task demands control acquisition and storage of visual information, *J. Exp. Psychol. Human* **31**, 1416–1438.

Fan, J., McCandliss, B. D., Fossella, J., Flombaum, J. I. and Posner, M. I. (2005). The activation of attentional networks, *NeuroImage* **26**, 471–479.

Firby, R. J., Kahn, R. E., Prokopowicz, P. N. and Swain, M. J. (1995). An architecture for vision and action, in: *Int. Joint Conf. on Artificial Intelligence*, pp. 72–79.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, USA.

Guestrin, C. E., Koller, D., Parr, R. and Venkataraman, S. (2003). Efficient solution algorithms for factored MDPs, *J. Artif. Intell. Res.* **19**, 399–468.

Hayhoe, M. M., Shrivastava, A., Mruczek, R. and Pelz, J. (2003). Visual memory and motor planning in a natural task, *J. Vision* **3**, 49–63.

Hikosaka, O., Bromberg-Martin, E., Hong, S. and Matsumoto, M. (2008). New insights on the subcortical representation of reward, *Curr. Opin. Neurobiol.* **18**, 203–208.

Humphrys, M. (1996). Action selection methods using reinforcement learning, in: *From Animals to Animats 4: Proc. 4th Internat. Conf. Simulation of Adaptive Behavior*, P. Maes, M. Mataric, J.-A. Meyer, J. Pollack and S. W. Wilson (Eds), MIT Press, Bradford Books, Cambridge, MA, USA, pp. 135–144.

Hurley, S. (2008). The shared circuits model (scm): how control, mirroring, and simulation can enable imitation, deliberation, and mindreading, *Behav. Brain Sci.* **31**, 1–22.

Itti, L. (2005). Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes, *Vis. Cogn.* **12**, 1093–1123.

Itti, L. and Koch, C. (2000). A saliency based search mechanism for overt and covert shifts of attention, *Vision Research* **40**, 1489–1506.

Karlsson, J. (1997). Learning to solve multiple goals, *PhD thesis*, University of Rochester, NY, USA.

Laird, J. E., Newell, A. and Rosenblum, P. S. (1987). Soar: an architecture for general intelligence, *Artif. Intell.* **33**, 1–64.

Langley, P. and Choi, D. (2006). Learning recursive control programs from problem solving, *J. Mach. Learn. Res.* **7**, 493–518.

Luck, S. J. and Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions, *Nature* **390**, 279–281.

Meuleau, N., Hauskrecht, M., Kim, K.-E., Peshkin, L., Kaelbling, L., Dean, T. and Boutilier, C. (1998). Solving very large weakly coupled Markov decision processes, in: *Proc. 15th Natl/10th Conf.* Madison, WI, USA, *AAAI/IAAI*, pp. 165–172.

Navalpakkam, V., Koch, C., Rangel, A. and Perona, P. (2010). Optimal reward harvesting in complex perceptual environments, *Proc. Nat. Acad. Sci. USA* **107**, 5232–5237.

Neisser, U. (1967). *Cognitive Psychology*. Appleton-Century-Crofts, New York.

Noe, A. (2005). *Action in Perception*. MIT Press, Cambridge, MA, USA.

Nordfang, M., Dyrholm, M. and Bundesen, C. (2012). Identifying bottom-up and top-down components of attentional weight by experimental analysis and computational modelling, *J. Exp. Psychol. Gen.*, DOI:10.1037/a0029631.

O'Regan, J. K. and Noe, A. (2001). A sensorimotor approach to vision and visual consciousness, *Behav. Brain Sci.* **24**, 939–973.

Parr, R. and Russell, S. (1997). Reinforcement learning with hierarchies of machines, in: *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns and S. A. Solla (Eds). MIT Press, Cambridge, MA, USA.

Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. Bradford Books, Cambridge, MA, USA.

Posner, M. I. and Rothbart, M. K. (2007). Research on attention networks as a model for the integration of psychological science, *Ann. Rev. Psychol.* **58**, 1–23.

Rao, R. P. N., Zelinsky, G. J., Hayhoe, M. M. and Ballard, D. H. (2002). Eye movements in iconic visual search, *Vision Research* **42**, 14447–14463.

Ritter, S., Anderson, J. R., Cytrynowicz, M. and Medvedeva, O. (1998). Authoring content in the pat algebra tutor, *J. Interact. Media Educ.* **98**, 1–30.

Roelfsema, P. R., Khayat, P. S. and Spekreijse, H. (2003). Sub-task sequencing in the primary visual cortex, *Proc. Nat. Acad. Sci. USA* **100**, 5467–5472.

Rothkopf, C. A. and Ballard, D. H. (2009). Image statistics at the point of gaze during human navigation, *Vis. Neurosci.* **26**, 81–92.

Rothkopf, C. A. and Ballard, D. H. (2010). Credit assignment in multiple goal embodied visuomotor behaviour, *Front. Psychol.* **1**, 1–13, online.

Rothkopf, C. A., Ballard, D. H. and Hayhoe, M. M. (2007). Task and context determine where you look, *J. Vision* **7**, 1–20.

Roy, D. K. and Pentland, A. P. (2002). Learning words from sights and sounds: a computational model, *Cogn. Sci.* **26**, 113–146.

Rummery, G. A. and Niranjan, M. (1994). Online Q-learning using connectionist systems, *Technical Report CUED/FINFENG/TR 166*, Cambridge University Engineering Department, UK.

Russell, S. and Zimdars, A. (2003). Q-decomposition for reinforcement learning agents, in: *Proc. Int. Conf. Machine Learning*.

Ruthruff, E., Pashler, H. E. and Hazeltine, E. (2003). Dual-task interference with equal task emphasis: graded capacity-sharing or central postponement? *Atten. Percept. Psycho.* **65**, 801–816.

Sallans, B. and Hinton, G. E. (2004). Reinforcement learning with factored states and actions, *J. Mach. Learn. Res.* **5**, 1063–1088.

Samejima, K., Doya, K. and Kawato, M. (2003). Inter-module credit assignment in modular reinforcement learning, *Neural Networks* **16**, 985–994.

Schultz, W. (2000). Multiple reward signals in the brain, *Nat. Rev. Neurosci.* **1**, 199–207.

Schultz, W., Dayan, P. and Montague, P. R. (1997). A neural substrate of prediction and reward, *Science* **275**, 1593–1599.

Shapiro, L. (2011). *Embodied Cognition*. Routledge, New York, USA.

Singh, S. and Cohn, D. (1998). How to dynamically merge Markov decision processes, in: *Neural Information Processing Systems Conf.*, Denver, CO, USA, 1997, Vol. 10, pp. 1057–1063.

Sprague, N. and Ballard, D. (2003). Multiple-goal reinforcement learning with modular sarsa(0), in: *Internat. Joint Conf. Artificial Intelligence*, Acapulco, USA.

Sprague, N., Ballard, D. and Robinson, A. (2007). Modeling embodied visual behaviors, *ACM Trans. Appl. Percept.* **4**, 11.

Stewart, J., Gapenne, O. and Di Paolo, E. (Eds) (2010). *En-action: Toward a New Paradigm for Cognitive Science*. MIT Press, Cambridge, MA, USA.

Sun, R. (2006). *Cognition and Multi-Agent Interaction*, Ch. 4, pp. 79–99. Cambridge University Press, UK.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.

Sutton, R. S., Precup, D. and Singh, S. P. (1999). Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning, *Artif. Intell.* **112**, 181–211.

Tolman, E. C. (1948). Cognitive maps in rats and men, *Psycholog. Rev.* **55**, 189–208.

Torralba, A., Oliva, A., Castelhano, M. and Henderson, J. M. (2006). Contextual guidance of attention in natural scenes: the role of global features on object search, *Psychol. Rev.* **113**, 766–786.

Treisman, A. M. (1980). A feature-integration theory of attention, *Cogn. Psychol.* **12**, 97–136.

Trick, L. M. and Pylyshyn, Z. W. (1994). Why are small and large numbers enumerated differently? A limited-capacity preattentive stage in vision, *Psychol. Rev.* **101**, 80–102.

Ullman, S. (1985). Visual routines, *Cognition* **18**, 97–159.

Vareala, F. J., Thompson, E. and Rosch, E. (1991). *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press, Cambridge, MA, USA.

Vigorito, C. M. and Barto, A. G. (2010). Intrinsically motivated hierarchical skill learning in structured environments, *IEEE Trans. Auton. Mental Dev.* **2**, 132–143.

Watkins, C. J. C. H. (1989). Learning from delayed rewards, *PhD thesis*, University of Cambridge, UK.

Yu, C. and Ballard, D. (2004). A multimodal learning interface for grounding spoken language in sensorimotor experience, *ACM Trans. Appl. Percept.* **1**, 57–80.

Appendix A: Reinforcement Learning Background

A standard formalism for describing the brain's programs is that of Markov Decision Processes (MDPs). An individual MDP consists of a 4-tuple (S, A, T, R) with S being the set of possible states, A the set of possible actions, T the transition model describing the probabilities $P(s_{t+1}|s_t, a_t)$ of reaching a state s_{t+1} when being in state s_t at time t and executing action a_t, and R is a reward model that describes the expected value of the reward r_t, which is distributed according to $P(r_t|s_t, a_t)$ and is associated with the transition from state s_t to some state s_{t+1} when executing action a_t.

The goal of RL is to find a policy π that maps from the set of states S to actions A so as to maximize the expected total discounted future reward through some form of learning. The dynamics of the environment T and the reward function R are not known in advance and an explicit reward function R is learned from experience. RL algorithms effectively assign a value $V^\pi(s)$ to each state, which represents this expected total discounted reward obtainable when starting from the particular state s and following the policy π thereafter. Where γ is a scalar factor that discounts future rewards, $V^\pi(s)$ can be described by:

$$V^\pi(s) = E^\pi\left(\sum_{t=0}^{\infty} \gamma^t r_t\right). \tag{8}$$

Alternatively, the values can be parametrized by state and action pairs, denoted by $Q^\pi(s, a)$. Where Q^* denotes the Q-value associated with the optimal policy π^*, the optimal achievable reward from a state s can be expressed as $V^*(s) = \max_a Q^*(s, a)$ and the Bellman optimality equations for the quality values can be formulated as in equation (9):

$$Q^*(s, a) = \sum_r r\,P(r|s, a) + \gamma \sum_{s' \in S} P\left(s'|s, a\right) \max_{a'} Q^*\left(s', a'\right). \tag{9}$$

Temporal difference learning (Sutton and Barto, 1998) uses the error between the current estimated values of states and the observed reward to drive learning. In its related Q-learning form, the estimate of the quality value of a state–action pair is adjusted by this error δ_Q using a learning rate α:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\delta_Q. \tag{10}$$

Two important expressions for δ_Q are: (1) the original Q-learning rule (Watkins, 1989) and (2) SARSA (Rummery and Niranjan, 1994). The Q-learning rule is an off-policy rule, i.e. it uses errors between current observations and estimates of the values for following an optimal policy, while

actually following a potentially suboptimal policy during learning. SARSA (see Note 2) is an on-policy learning rule, i.e. the updates of the state and action values reflect the current policy derived from these value estimates. As SARSA allows one to follow a sub-optimal policy in the course of learning, it is well-matched for use with modules, which cannot always depend on following their own policy recommendations. Its learning rule is given by:

$$\delta_Q = r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t). \tag{11}$$

Evidence that both the Q-learning and the SARSA error signals are represented in the brain of animals and humans have been provided in numerous experiments e.g. (Schultz, 2000; Schultz *et al.*, 1997a).

Appendix B: Embodied Module Definitions

We assume that the required behaviors can be realized with separate RL modules. The primary assumption is that such modules are activated in subsets whose members do not interfere with each other (Guestrin *et al.*, 2003; Russell and Zimdars, 2003; Sprague *et al.*, 2007).

An independent RL module with its own actions can be defined in different ways, but the formalism here defines a module as an MDP i.e. the ith module is given by (see Appendix A for definitions of notation):

$$\mathcal{M}_i = \{S_i, A_i, T_i, R_i\}, \tag{12}$$

where the subscripts denote that the information is from the ith MDP.

The states of the different modules are assumed all non-overlapping. In such a case, the optimal value function is readily expressible in terms of the component value functions and the states and actions are fully factored so that there is no overlap between states and additionally the conditional probabilities for the state and reward at time $t + 1$ are simply the product of the constituent values in the individual modules. Where $\mathbf{s} = \{s^{(1)}, \ldots, s^{(M)}\}$ is the combined state of the M modules and similar notation is used for $\mathbf{a}$ and $\mathbf{r}$,

$$P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) = \prod_{i=1}^{M} P\left(s_{t+1}^{(i)}|s_t^{(i)}, a_t^{(i)}\right), \tag{13}$$

$$P(\mathbf{r}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) = \prod_{i=1}^{M} P\left(r_{t+1}^{(i)}|s_t^{(i)}, a_t^{(i)}\right). \tag{14}$$

These two conditions can be used together with equation (9) in order to arrive at the result:

$$Q(s_t, a_t) = \sum_{i=1}^{N} Q\left(s_t^{(i)}, a_t^{(i)}\right). \tag{15}$$

If equations (14) and (13) hold and all the rewards are known, the action maximizing equation (15) can be selected and is guaranteed to be optimal. In this decomposed formulation, each module can follow its own policy π^i, mapping from the local states s^i to the local actions a^i. This case is appropriate for the multi-agent setting when each module can be identified with an agent that may be expected to act independently.

However, in the embodied cognition setting that is our focus, a single agent pursues multiple goals that are divided up between multiple independent modules that the agent can activate concurrently (Humphrys, 1996; Karlsson, 1997; Singh and Cohn, 1998; Sprague and Ballard, 2003). The main problem this concurrency introduces is that the action space is shared, so the ith module definition is now:

$$\mathcal{M}_i = \{S_i, A, T_i, R_i\}, \tag{16}$$

where all active modules use the same action set A. Thus the embodiment requires some form of action selection in order to mediate the competition between the possibly rivalrous actions proposed by individual modules. We use the probabilistic softmax action selection:

$$P\left(a_t^{(j)} \middle| Q\left(s_t^{(1)}, a_t\right), \ldots, Q\left(s_t^{(N)}, a_t\right)\right) = \frac{e^{Q(s_t^{(j)}, a_t^{(j)})/\tau}}{\sum_{i=1}^{M} e^{Q(s_t^{(i)}, a_t^{(i)})/\tau}}. \tag{17}$$

To choose the action, and once it has been selected, it is used for all modules. This type of action selection has been shown to model human behavior well in a variety of decision making tasks (Daw *et al.*, 2006; Navalpakkam *et al.*, 2010). The parameter τ controls the balance between exploration and exploitation during learning and usually decreases over time to reflect the shift toward less exploratory decisions over the course of learning.

This model has been very effective in representing human performance in the case where the multiple tasks are to walk down a sidewalk while simultaneously staying on the sidewalk, picking up litter objects and avoiding obstacles. Some of the gaze data for humans performing this task was reviewed in Fig. 3. Figure 10, from Sprague *et al.* (2007), shows the results of the learning *via* RL of separate modules for each of these three tasks by an avatar model embedded in the same environment. The top panels in the figure show the discounted reward values as a function of the state space in front of the agent. The bottom panels show the respective policies. Note that for each of the modules the state estimate is different, as a consequence of the disposition of the agent in the environment and the relative positions of surrounding objects. Figure 10 illustrates the action selection issue that crops up with the use of modules: actions recommended by individual modules may be different. For the walking environment it suffices to take the reward-weighted average of the Q values of the

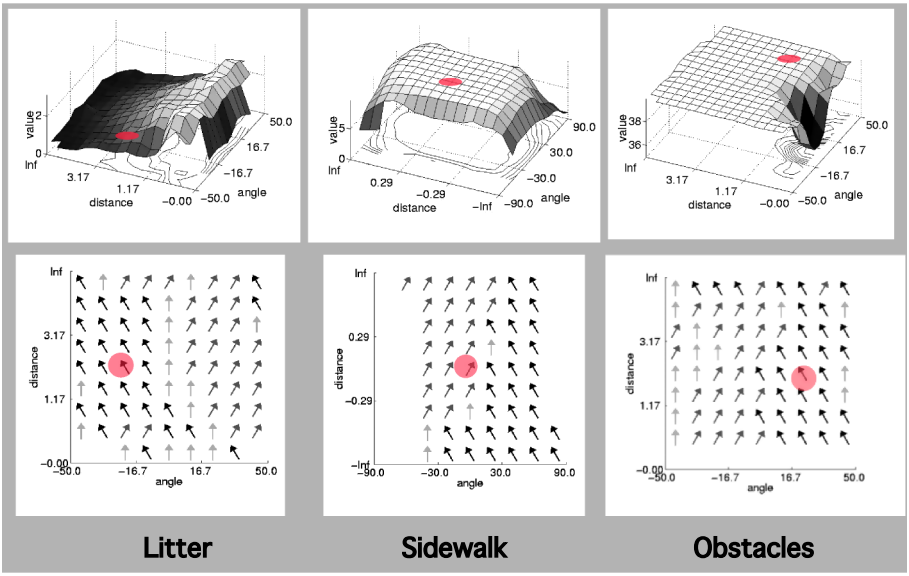

Figure 10. Value functions and their associated policies for each of three modules that have been learned by a virtual avatar walking along a sidewalk strewn with litter and obstacles. The red disk marks the state estimate for each of them. The individual states for each module are assumed to be estimated by separate applications of the gaze vector to compute the requisite information. Thus the state for the obstacle is the heading to it, and similarly for the state for a litter object. The state for the sidewalk is a measure of the distance to its edge. In the absence of a gaze update, it is assumed that subjects use vestibular and proprioceptive information to update the individual module states. This figure is published in color in the online version.

possible actions (equation (17)), but in general, more sophisticated techniques may be used.