BIROn - Birkbeck Institutional Research Online

The Goal Circuit Model: A hierarchical multi-route model of the acquisition and control of routine sequential action in humans[*]

Richard P. Cooper
Nicolas Ruh
Denis Mareschal

Centre for Cognitive and Computational Modeling
Department of Psychological Sciences, Birkbeck, University of London

Abstract: Human control of action in routine situations involves a flexible interplay between (a) task dependent serial ordering constraints, (b) top-down, or intentional, control processes and (c) bottom-up, or environmentally-triggered, affordances. Additionally, the interaction between these influences is modulated by learning mechanisms that, over time, appear to reduce the need for top-down control processes while still allowing those processes to intervene at any point if necessary or if desired. We present a model of the acquisition and control of goal-directed action that goes beyond existing models by operationalizing an interface between two putative systems – a routine and a non-routine system – thereby demonstrating how explicitly represented goals can interact with the emergent task representations that develop through learning in the routine system. The gradual emergence of task representations offers an explanation for the transfer of control with experience from the non-routine goal-based system to the routine system. At the same time it allows action selection to be sensitive both to environmental triggers and to biasing from multiple levels within the goal system.

Keywords: routine sequential action, control of action, habits, goal-directed action, purposive-action, hierarchical tasks

Introduction

Eight characteristics of the control of sequential action

The execution and control of routine sequential action is a core component of everyday human activities. In highly over-learned tasks the control system (or systems) responsible is capable of functioning with only occasional errors (Reason, 1979), even when simultaneously carrying out other, apparently cognitively demanding, tasks. For example, it is usually possible to conduct an engaging conversation while completing a routine task such as preparing a mug of coffee. More specifically, the control of sequential action in humans can be characterized by the following eight properties.

1. Sequential Action is Purposive (i.e., Goal-Directed). Consider an action sequence such as: ‹‹*open sugar bowl, pick up spoon, dip spoon in sugar bowl, empty spoon into beverage, stir beverage, discard spoon*››. This sequence will achieve the goal of sweetening a beverage (amongst other things), and if an agent's goal was to sweeten a beverage then a plausible approach would be for the agent to attempt to perform this (or some closely related) sequence of actions.

2. Action is Hierarchically Structured. The sequence of actions involved in everyday tasks, such as preparing a cup of tea, may be decomposed into a sequence of subsequences, where the subsequences cohere because they achieve subgoals of the original task (such as boiling

---

[*] Please direct correspondence to Dr. Richard P. Cooper, Department of Psychological Sciences, Birkbeck, University of London, Malet Street, London WC1E 7HX. Email: R.Cooper@bbk.ac.uk

the water or sweetening the tea) and because they may be used in different tasks (such as preparing a mug of instant coffee). Transitions from one subsequence into the next are, conversely, less coherent and more susceptible to error (e.g., Reason, 1979; Botvinick & Bylsma, 2005), presumably because other tasks might require different transitions. Action thus has a hierarchical structure, and this structure reflects the goal/subgoal structure of the action sequence's highest-level goal.

3. Actions within a Sequence are not all Equal. While action sequences are hierarchically structured, there is some flexibility in that structure. This flexibility relates to the different functions that actions serve within a sequence. In particular, "enabling" actions (e.g., picking up a spoon to stir a beverage) and "clean-up" actions (e.g., discarding the spoon once the beverage has been stirred) may be omitted in any specific instance if they are redundant (Schwartz et al., 1991). Thus, sweetening a beverage with two sugars does not involve performing the beverage sweetening sequence in its entirety twice. Rather, the two sequences are normally run together (without discarding the spoon and picking it up again between the two subsequences) and this running together occurs with little or no apparent cognitive cost or load.

4. Action may be Controlled at Multiple Levels. Consider an everyday task such as dressing or grooming. Phenomenologically, one might perform such a task as a series of subtasks (e.g., fetching a pair of socks, and then putting a sock on each foot). Alternatively, one may deliberately perform the sub-ordinate acts within each subtask (walking to the sock drawer, opening the drawer, etc.). In the case of well-learned actions, it may even be possible to perform the super-ordinate task without deliberately attending to the individual subtasks (Norman, 1981). Consider the situation when entering a computer password. Phenomenologically, it is possible to do this without focusing on the individual characters. However, if errors are made on the original attempt through, for example, excess haste, and one needs to renter the password correctly or risk being locked out of ones account, then it is possible to exert control at the level of individual actions. Thus, and as argued by Heckhausen and Beckmann (1990, p. 36) when considering purposive action, "the action-guiding intention can be identified at various goal levels".

5. Action may be Triggered by Features of the Environment. As reviewed by Riddoch, Edwards, Humphreys, West and Heafield (1998), a substantial body of evidence from behavioral and neuropsychological studies strongly supports the view that actions may be primed or triggered by features of the environment. For example, in diary studies of slips and lapses in everyday action, Reason (1979, 1984) and Norman (1981) found numerous reports of action errors in which an unintended but environmentally appropriate action was performed. Such errors are more common (and more flagrant) in neuropsychological disorders of action selection, such as utilization behavior (where patients are prone to spontaneously use objects present in their immediate environment in object-appropriate ways: Lhermitte, 1983; Shallice, Burgess, Shon & Baxter, 1989), anarchic hand syndrome (where one of a patient's hands acts independently of, and often in conflict with, the patient's stated intentions: see, e.g., Della Sala, Marchetti, & Spinnler, 1991; Goldberg, Mayer, & Toglia, 1981) and action disorganization syndrome (where the integrity of goal-directed action is compromised, with action sequences including frequent omission, action addition, and object substitution errors: Schwartz et al., 1991, 1998; see also Luria, 1966; Duncan, 1986).

6. Routine Action Requires Minimal Attentional Control. One key finding from the diary studies of Reason (1979, 1984) and Norman (1981) is that action in routine situations is prone to error when attention is diverted by some other task (see also Heckhausen & Beckmann,

1990). Viewing this finding from an alternative perspective, it is clear that while action slips and lapses may be relatively common when attention is not directed to a routine task, they are by no means universal.[1] Thus, in many situations it is possible to perform routine action sequences without overt attention and without error (James, 1890; Norman & Shallice, 1986). For example, it is generally possible to listen to the morning news while making coffee without error, or to put on socks and shoes while having a conversation.

7. Learning Reduces the Need for Attentional Control. An obvious implication of the previous property together with the fact that attentional control is required for non-routine action (or action in non-routine situations) is that the need for attentional control is reduced by learning or practice. That is, a consequence of repeated performance of an action sequence in a specific situation is that the action sequence comes to be automatized. If we accept that prefrontal cortex (PFC) is involved in the attentional control of goal-directed behavior (e.g. Miller & Cohen, 2001), then the decreasing need for attention with practice is consistent with neuroimaging studies (e.g., Jenkins, Brooks, Nixon, Frackowiak, & Passingham, 1994; Jueptner et al., 1997; Passingham, Rowe & Sakai, 2005) which show activation of regions of the PFC to be greater during the early stages of acquisition of a task, and lesion studies (e.g., Beldarrain, Grafman, Pascual-Leone, & Garcia-Monco, 1999; Richer, Chouinard, & Rouleau, 1999) which show that procedural and motor learning are impaired by frontal lesions.

8. Attentional Control may Initiate and/or Override Learned Action Sequences. The final property of action control with which we are concerned is that while well-learned action sequences may be performed in routine situations with minimal attentional control, such control may nevertheless be used to override a learned sequence when necessary (e.g., when recovering from error) or desired (e.g., when trying to break a habit: Wood & Neal, 2007).

The "Dual Systems" theory of the control of sequential action

One widely accepted view of the control of sequential behavior which provides contact with these eight properties is Norman and Shallice's (1986) "Dual Systems" theory. Within this theory, sequential action is controlled by Contention Scheduling (CS), a conflict resolution system that requires minimal cognitive resources. This system is held to select from amongst the myriad of actions possible at any moment in time, and be capable of functioning autonomously during the control of routine behavior. Importantly, CS is held to trade-off between two distinct influences: bottom-up environmental triggers, sometimes termed 'exogenous control' on the one hand, and task-specific ordering constraints (so-called 'horizontal threads') on the other hand. However, in non-routine or deliberative behavior (e.g., less familiar circumstances, novel tasks or dangerous situations), a third influence might come into play: an executive system, the Supervisory System (SS), may modulate CS via top-down control (so-called 'vertical threads'). The relationship between these two systems (CS and SS) may be likened to the relationship between a horse and a rider:

a.   the horse is the one who actually carries out the work (of locomotion), and it takes environmental constraints into account when doing this, e.g., not walking into obstacles

b.   the rider may bias the horse at problematic points, e.g., in choosing to go left or right at a crossing

---

[1] We are concerned here with attention in the sense of "attention to action", following Norman and Shallice (1986; see also Humphreys & Riddoch, 2005), and not "visual attention". Attention to action corresponds phenomenologically to cognitive effort when selecting actions.

    c.   the rider is not necessarily more intelligent than the horse (and is thus not a homunculus) – he would not know which leg to move when – but he might be better at deciding upon an appropriate path, especially in unfamiliar or ambiguous circumstances

    d.   in familiar situations the horse can do well without any influence from the rider (e.g., the horse is able to find its way home on its own along a familiar route)

As the horse and rider analogy makes clear, while the theory posits two different systems that contribute to the control of action, it is not claimed that action is controlled by switching between one system and the other. Rather, action selection is the product of CS (i.e., the horse) acting under occasional or sporadic bias from SS (i.e., the rider). In routine situations, CS may function appropriately in the absence of input from SS (although redundant input from SS should not pose a problem), while in non-routine situations input from SS is required to modulate routine CS functioning in order to produce appropriate or desired behavior.

At an informal level, the Dual Systems view is consistent with each of the properties described above (see, e.g., Norman & Shallice, 1986; Shallice, 1988). It is also consistent with work in the animal psychology literature, which argues that habits and goal-directed actions are affected differentially by reward (e.g., Dickinson, 1985; see also Dickinson & Balleine, 2002). Shallice (2006) provides a further argument for distinct systems for the control of routine and non-routine action, namely that the systems are sensitive to different variables. In the case of routine action, Shallice argues that the key variables affecting performance are those of familiarity, age-of-acquisition, and frequency of application. These variables, Shallice notes, are ones typically linked with associationist (or, in contemporary computational terms, connectionist) theories. Non-routine behavior, or deliberative control, does not appear to be affected by these variables.

If we are to take the implications of the Dual Systems theory seriously, a clear account of the interplay between CS and SS – the harness or reins, if you like, used by the rider to control the horse – is needed to help understand how sequential action is governed. These two systems need to be able to interact in a flexible way in order to satisfy the properties enumerated above. Thus, SS must be able to support CS efficiently when required at problematic points in a sequence, while, at the same time, CS must strive for independence so as to be able to work autonomously in sufficiently well-learned situations. Of course, deciding what makes a specific choice 'sufficiently well-learned' is a significant challenge – firstly, because the amount of experience with a specific sequence and the choices to be made during its execution will change over time, and secondly, because the level of selection difficulty at any one point in a sequence might furthermore depend on factors such as the number of, and familiarity with, other locally similar sequences, the number of alternative ways of achieving the current goal, the temporal distance between co-dependent actions, and the presence or absence of perceptual disambiguation cues in the environment. Empirical work has shown that all of these factors can exert an influence on the time between selection of actions within a semi-routine goal-directed sequence (Ruh et al., 2010).

Existing Models of the Control of Sequential Action

To date, there are three main computational accounts of routine sequential action selection within the psychological literature. One key difference between the accounts concerns their underlying task representation. One, the simple recurrent network (SRN) model of Botvinick and Plaut (2004), employs emergent, distributed representations in conjunction with a (temporal) context layer to control task execution. The second, the interactive activation

network (IAN) model of Cooper and Shallice (2000) relies on explicit, hierarchically structured task representations or schemas, where subgoals mediate schema-subschema associations. The third, the "memory for goals" (MfG) model of Trafton et al. (2011; see also Altmann & Trafton, 2002), assumes that action is guided by what they call episodic control codes that are subject to decay but may be primed by context. These control codes may be hierarchically structured, reflecting task structure. The SRN model, due to its associationist/connectionist origins, is sensitive to precisely the variables identified by Shallice (2006) as affecting routine behavior. For example, the errors made by the model are determined by the sequences on which it was trained (i.e., familiarity). The IAN and MfG models, conversely, capture more of the goal-directed nature of sequential action and allow for the inclusion of, for example, monitoring and error correction mechanisms. (See Ruh, 2007, for a more detailed comparison and criticism of the SRN and IAN approaches.) The most important shortcoming of all models, however, is that they are models of CS only.[2] Therefore, they are unable to make any claims about behavior that is not entirely routinized, or indeed about the acquisition of routine tasks which, following our list of characteristic properties of human sequential action, involves a progressive transfer of control from higher level systems (SS) to lower level (CS) ones.

Two further existing models address this issue in different ways. Daw, Niv and Dayan (2005) present a mathematical account of sequential behavioral control based on two systems – one (corresponding in functionality to SS) is held to implement "tree-search" control and be associated with prefrontal cortex, while the other (corresponding in functionality to CS) is held to implement "cached" control and be associated with the dorsolateral striatum. Each system is held to learn through a form of Reinforcement Learning (RL: Sutton & Barto, 1998), but the prefrontal system is held to use RL to acquire a model of the environment which may be used for planning and prediction (so-called "model-based" RL) while the striatal system is held to use simpler "model-free" RL algorithms, which associate rewards with sequences of actions. Daw et al. argue that competition between these two systems is resolved an uncertainty-based controller that seeks to select the system whose output is expected to be most accurate. While this account has many strengths, the idea of uncertainty-based selection between the outputs of the two systems (rather than that one system – presumably the frontal system – might operate indirectly by biasing the other system) does not easily account for some key characteristics of sequential action, such as that action may be controlled at multiple levels (characteristic 4), and that attentional control may initiate and/or override learned action sequences (characteristic 8).

A second line of work where this transfer has been explored is in the context of skill acquisition. The model of skill acquisition of Taatgen et al. (2008), in particular, attempts to balance what the authors refer to as "top-down control" and "bottom-up control" so as to acquire skill knowledge that is both robust (e.g., to interruptions) and flexible (i.e., that generalize to related problems). These are clearly critical features of both acquired skills and everyday action. (See Cooper et al., 2005, for a development of the IAN model of CS that addresses these requirements.) Not withstanding subtle differences between the domains of cognitive skills and routine action, the current work shares goals with that of Taatgen et al. However, in contrast to those authors, whose model is developed within the ACT-R

---

[2] Botvinick and Plaut (2004) do not describe their SRN model in terms of the dual-ssystems model discussed here. However, they do consider the model to be one of routine sequential action, and not one of action more generally. That their model is intended specifically to be one of CS is clarified in their later work, where they relate their model to a so-called habit system, and "concur with Cooper and Shallice [2006] on the distinction between two interacting systems underlying action control, a habit (or Contention Scheduling) system and a goal-directed (or Supervisory Attentional) system" (Botvinick & Plaut, 2006, p. 923).
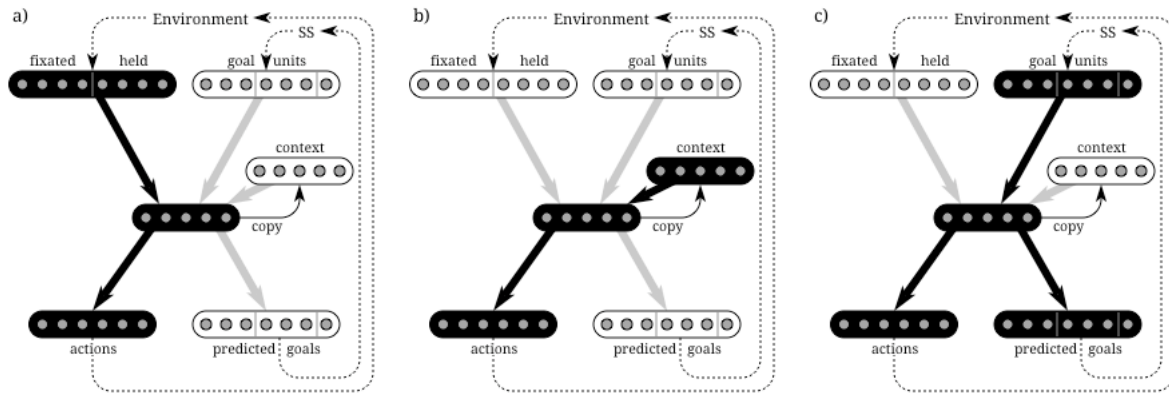
Figure 1: The abstract architecture of the GC model and the contributions of different routes to action selection within it: a) direct pathway; b) context pathway; c) goal circuit.

architecture, the model proposed below builds on the Parallel Distributed Processing approach to cognitive modeling. Specifically, our model attempts to combine the strengths of the SRN and IAN approaches (and representational formats), thereby allowing us to address the full range of influences proposed within the Dual Systems framework. We begin by describing our model, the Goal Circuit (GC) model, initially in abstract terms and then as applied in a concrete domain – that of beverage preparation. Three simulation studies are then presented which demonstrate how the model captures the characteristics of action enumerated above. Specifically, it is demonstrated that the GC model can replicate critical aspects of performance of the SRN model, while at the same time it is robust to noise and instructable through its goal-node interface – a key property of the IAN model. The general discussion then focuses on insights from the model and its relation to the wider literature.

The Goal Circuit Model: Assumptions and Architecture

Learning routine sequential action in a way that allows the flexible application of control when it is, and is not, necessary, remains a significant difficulty. Part of that difficulty lies in the fact that the influence of SS required by CS to function without error should decrease as CS gains experience on a task. This requirement conflicts with associationist approaches to learning where, if SS input were present and appropriate, learning would strengthen connections between SS and CS, thereby making CS more rather than less reliant on SS. This and the following section therefore present a dual-system model of action control in which the routine system gradually becomes capable of autonomous routine behavior. At the same time, the routine system can be biased by the non-routine system to produce non-routine behavior if necessary. We refer to the model as the Goal Circuit (GC) model. It extends Botvinick and Plaut's SRN model in three ways. Specifically, it provides:

   a.   a way to interface the model with an 'executive system' (SS) which can add top-down control (bias) in cases when action is not fully routinized;
   b.   a more plausible training regime, taking into account reusability of existing (sub)sequence knowledge and progressive reduction of executive control with increasing practice; and
   c.   the ability to reach the (sub)goal of a specific (sub)task in a flexible manner, thus dealing with minor variations in the states of objects (e.g., when preparing a beverage whether condiment packets are closed or already open) or the initial state of the system.

The Abstract Architecture

The abstract architecture of the GC model is illustrated in Figure 1. As in the original SRN of Botvinick and Plaut (2004), our model takes inputs representing the object on which the model is currently "fixated" and the object (if any) which is currently "held" by the model. Unlike the original SRN, the GC model also takes as input a representation of the current goal(s). The GC model aims to generate from these inputs an action and a representation of the subsequent or predicted goal(s). Actions may include moving fixation to another object, picking up the fixated object, or manipulating the held object in one of several ways (including putting it down). Representations of goals are fed back into the model on the next processing step, possibly modulated by an SS component. Input units feed through a hidden layer with recurrent connections which allow the model to develop an implicit representation of task context through learning. This context, in conjunction with the input at any step in a task, allows the model, once trained, to correctly generate the appropriate output and the context for the next step.

We take the Botvinick and Plaut model as our starting point because an SRN provides a natural implementation of both learning and sequence execution, and, as noted above, the variables to which routine action selection are sensitive (recency, frequency, etc.) are most easily captured by associationist learning principles (Shallice, 2006). However, in adopting a recurrent architecture we do not subscribe to the specific learning regime employed by Botvinick and Plaut (2004), or to the implicit, non-hierarchical (and hence non-compositional) and non-teleological schema representations that result from this regime (see Cooper & Shallice, 2006). Rather, two substantive augmentations and modifications to the basic SRN yield schema representations that implicitly encode hierarchy and purpose. First, an additional input to the model is provided by a bank of goal units. Second, and as a consequence, an additional route exist between perception and action, mediated by goals and paralleling the involvement of goals in Cooper and Shallice's IAN model.

Assumption 1: Multiple Routes in the Selection of Action

The GC model assumes that the selection of each individual action is the product of multiple influences or biases. Three specific influences are implemented. First, following characteristic 5, it is assumed that one input to the action selection mechanism is the representation of the current external environment (see Figure 1a). This route reflects Gibson's notion of direct affordances of objects for actions (Gibson, 1977). A second source of influence is, as described above, a sequencing system for the generation and regulation of well-learned or routine action sequences (CS; see Figure 1b). Finally, the hypothesized SS provides a third source of bias (see Figure 1c). This allows specific intentions to affect behavior (characteristic 8), including predicted goals produced by processing of the network on the previous processing step. We view the three sources of bias as different routes from perception to action. Importantly, the routes are not mutually exclusive. Rather, they are held to operate in conjunction in the functioning model. Action selection at any moment therefore reflects the superposition of the influences of the various routes. Note that the three routes correspond approximately to the three different influences (bottom-up environmental triggers, horizontal ordering constraints and top-down control) that the Dual Systems framework posits to interact within Contention Scheduling.

Assumption 2: Goal Units and Goal-Based Learning

The second assumption of the GC model (following properties 1, 2 and 3) is that goals are encoded at multiple levels within the action selection system(s) and these encodings may bias action selection at any point. For current purposes, goals are encoded using a localist

representation (e.g., one node represents the goal of making coffee, while another represents the sub-goal of adding sugar, etc.). The assumption is that these nodes may be indirectly activated by an (external) executive system, or directly activated through the automatic feedback of predicted goals (dotted lines in Figure 1 from predicted goals via SS to goal units; see below for more detail).

The inclusion of goal units allows for what we consider to be a more psychologically plausible approach to learning. Thus, within the GC model learning is goal-based rather than sequence-based. The original SRN model of Botvinick and Plaut (2004) was trained on six specific sequences corresponding to two tasks (four sequences for preparing a mug of instant coffee, and two for preparing a mug of tea), and was able to reproduce these six sequences. Learning in the GC model instead involves encoding sequences that achieve task goals and sub-goals. Thus, for each goal / sub-goal of a task, the model is trained to achieve the goal / sub-goal from any given state of the environment. While both models use the same underlying learning algorithm (back-propagation through time), it will be shown the emphasis on goal-based sequence learning within the GC model results in a system that is both robust and flexible.

A corollary of the incorporation of goal units and goal-based learning is that if a network has already learned action subsequences (such as how to add coffee grounds, sugar, cream, etc. when making a beverage) then it should be possible to guide the model to complete more complex or novel sequences (e.g., making a cup of coffee) made up of the individual subsequences by activating goal units associated with each subsequence in the correct order. At the same time sufficient experience of the transitions between subsequences required for a complex sequence should result in knowledge of subsequence ordering becoming associated with higher-level goal units, thus making the top-down guidance at the lower level optional. The same argument applies to the lower-level goals (e.g., acquisition of an "add sugar to beverage" routine), down to basic schemas (e.g., pick something up), where ordering is fully determined by environmental constraints.

Comparison with the SRN Model of Botvinick and Plaut (2004)

To summarize, while the GC architecture builds upon Botvinick and Plaut's SRN architecture, it differs in the two critical ways. First, the network is augmented with a "goal circuit", which contains explicitly represented goals that are predicted from and fed back into the shared hidden layer, mediated through the hypothesized SS. The goal input bank corresponds to the interface with SS. Units here are used to exert top-down control over the basic CS system (i.e., the core SRN of Botvinick & Plaut, 2004). Second, the network is trained not with specific sequences but with specific goals, and moreover it is trained with goals at all levels of the hierarchy. That is, the network is trained not merely to make coffee by following a rote sequence, but also to add sugar to a beverage, and to get sugar, open it, etc. This offers the possibility of applying relevant subsequence in novel tasks or novel environments.

The Beverage Preparation Task

In order to fully specify the abstract GC model it is necessary to commit to a particular task and a specific input/output representation. Two of the previous computational accounts of routine sequential action described above (i.e., Cooper & Shallice, 2000; Botvinick & Plaut, 2004) have been expressed as models of behavior on the routine task of preparing a cup of tea or instant coffee. This task, which has been the subject of several empirical studies (e.g., Giovannetti, Schwartz & Buxbaum, 2007; Humphreys & Forde, 1998; Humphreys, Forde &

Francis, 2000; Reason, 1990; Ruh et al., 2010; Schwartz et al., 1998; Schwartz, Reed, Montgomery, Palmer, & Mayer, 1991; Land, Mennie & Rusted, 1999), captures several challenging aspects of hierarchical sequential tasks, such as having the same subsequence appear in different tasks (e.g., sugar may be added to either tea or coffee) and/or in different orders (e.g., cream may be added before or after sugar when making coffee), and with different versions of subsequences (e.g., using sugar from a pack or from a bowl) being interchangeable. For these reasons it is the task that we adopt here. More precisely, we simulate the acquisition of beverage preparation as described by Botvinick and Plaut (2004), though we characterize the domain is in terms of its goal/subgoal structure. Thus, we assume 18 goals at three levels which decompose (eventually) into 19 actions (see Table 1):

- 2 high-level goals: *make tea, make coffee*

- 5 mid-level goals: *add grounds, add teabag, add sugar, add cream, drink*

- 11 low-level goals: *get object* (for each of seven objects)*, open, add, stir, sip*

Again, following Botvinick and Plaut (2004), there is one way to make tea, and two ways of making coffee (depending on the ordering of the creaming and sugaring subtasks):

> *Make tea ← add teabag, add sugar, drink*
>
> *Make coffee ← add grounds, add sugar, add cream, drink*
>
> *Make coffee ← add grounds, add cream, add sugar, drink*

The five mid-level goals similarly decompose into sequences of low-level goals:

> *Add grounds ← get-coffee-pack, open, add, stir*
>
> *Add teabag ← get-teabag, dip*
>
> *Add sugar ← get-sugar, [open,] add, stir*
>
> *Add cream ← get-cream-carton, open, add, stir*
>
> *Drink ← get-mug, sip, sip*

As noted above, the model takes as input a representation of a held object and a representation of a fixated object, from which it generates an action such as *put down* (the held object) or *pick up* (the fixated object). The decomposition of low-level goals into basic actions depends upon the fixated and held objects, so that a low-level goal such as *get-cream-carton* might be achieved by anything between zero and three actions. If the cream carton is already held, then no actions are required, but if some other object is being held and the cream carton is not fixated, then three actions will be necessary: *put-down* (whatever is being held), *fixate-cream-carton*, and *pick-up* (whatever is being fixated, i.e., the cream carton).

For simulations 1 and 2 below it is assumed that at the beginning of the task the state of the world is such that nothing is fixated, nothing is held, and all containers are sealed. In this situation most low-level goals correspond to a fixed sequence of basic actions. The only exceptions relate to sub-goals of *add sugar*. Following Botvinick and Plaut (2004), two sugar sources are included in the environment, and the consequence of performing *get-sugar* depend upon which happens to be returned by the *fixate-sugar* action. If this action results in the (sealed) sugar packet being the fixated object, then the appropriate subsequent actions are to tear the packet open and add its contents to the mug. (This involves fixating upon the mug and pouring the content of the held object into the mug.) If, however, *fixate-sugar* results in the sugar-bowl being the fixated object, then *get-sugar* must also open the sugar-bowl by *pulling-off* its lid and putting it down, *get* the teaspoon, and *scoop* from the sugar-bowl with the teaspoon.

Input unit	Object fixated or held	Output unit	Action
1	cup	1	pick-up
2	1-handle	2	put-down
3	2-handles	3	pour
4	lid	4	peel-open
5	clear-liquid	5	tear-open
6	light	6	pull-open
7	brown-liquid	7	pull-off
8	carton	8	scoop
9	open	9	sip
10	closed	10	stir
11	packet	11	dip
12	foil	12	say-done
13	paper	13	fixate-cup
14	torn	14	fixate-teabag
15	untorn	15	fixate-coffee-pack
16	spoon	16	fixate-spoon
17	teabag	17	fixate-carton
18	sugar	18	fixate-sugar
19	nothing	19	fixate-sugar-bowl

Table 1: Details of the model's input and output representations: Following Botvinick and Plaut (2004), we use 18 units to encode fixated objects and 19 to encode held objects, with the 19th held unit encoding that nothing is held. Individual objects are coded by up to five units, for example activating nodes 1, 2 and 5 represents a cup filled with water.

Simulation 3 considers the more general situation where the initial state of the world varies from trial to trial and so where containers may or may not be open. As should be clear from this discussion, the characterization of the domain in terms of goals and subgoals allows for subgoals which may in some situations be unnecessary and hence which in some situations should be omitted.

Network Architecture for the Beverage Preparation Task

In order to model beverage preparation, the details of the GC model's architecture and parameters were held as close as possible to those of the original SRN model of Botvinick and Plaut (2004). Thus, a standard sigmoidal activation function (ranging from 0 to 1) was employed for all units of the network. The perceptual input consisted of 37 units coding the objects currently fixated (18 units) and held (19 units), while the output layer coded Botvinick and Plaut's (2004) 19 possible actions in a localist manner. (See Table 1 for details.) The goal layers (for both input goals and predicted goals) consisted of 18 units encoding the goals described above. The hidden layer had 50 units.

Simulation Studies

Simulation 1: Learning with Variable Direction from Goal Units

Before attempting to explore the intentional control of behavior within the GC model or the ability of the model to transfer learned behavior to novel situations, it is essential to demonstrate that the basic model is indeed capable of learning tasks of similar complexity to those employed in earlier work, and moreover that when trained the model is capable of

exploiting the goal circuit so as to function with different levels of guidance from goal input units, reflecting the fact that, once routinized, sequential action may be performed with or without over attention (characteristics 4, 6 and 7, from the introduction). This was the purpose of simulation 1.

Method

As discussed above, simple recurrent network models are typically trained with a closed set of target sequences, constructed prior to training and subject to the requirements of the specific task. In contrast, for the GC model target behaviors are defined in terms of goals as described above. Target action sequences are then constructed on the fly (i.e., during a trial and following each action) in the context of the changing state of a simulated world. To illustrate, suppose the target goal is to add sugar to a container. As discussed above, the sequence needed to achieve this goal will depend upon what forms of sugar are available in the immediate environment (e.g., a sugar packet versus a sugar bowl), and whether they are open or closed. Even if the initial state of the world is fixed, different sequences may be generated for different attempts at the same goal if individual steps are non-deterministic, as is the case in the current domain, where (as described above and following Botvinick & Plaut, 2004) the *fixate-sugar* action can result in fixation upon either the sugar packet or the sugar bowl. Given this approach, one training epoch was defined as one attempt at each of the 18 goals (the 11 low-level goals, the 5 mid-level goals, and the 2 high-level goals), with one sequence for each goal generated on the fly during each training trial from the changing representation of the state of the environment.

The model was trained using epoch-wise back-propagation through time with a cross-entropy error function (Williams & Zipser, 1995) and with a learning rate of 0.001, weight persistence of 0.999999, and weight updates applied after each item. Informal exploration suggested that with these settings training generally required 10,000 to 20,000 epochs, but that even then the model sometimes failed to learn the most complex sequences (those for making coffee). To explore training effects systematically, model behavior was therefore examined with four level of training: after 5,000 epochs, 10,000 epochs, 20,000 epochs and 40,000 epochs. In order to allow the network to incorporate optional dependence on the goal circuit, goal inputs were set to zero following the first step on half of the training epochs. On the other half, goal inputs were set to values corresponding to the current goal and subgoals, i.e., the values that a fully trained network should have generated as the predicted goal activations on the previous cycle.

For each training level, model weights were initially randomized to values uniformly distributed between –0.15 and +0.15 and the model was trained on the full goal-set for the specified number of epochs. The trained model was then tested one hundred times on each of the eighteen goals. Each test involved setting the values of all hidden units to random values uniformly distributed between 0.01 and 0.99, initializing the state of the environment, setting the relevant goal and state input unit(s), and then cycling through actions until either one hundred actions were attempted or the *say-done* action was selected. Sequences thus generated were scored as correct if they both ended with the *say-done* action and resulted in a state in which the initial goal was achieved.

In order to explore the effects of different levels of guidance, each trained network was tested in this way for each of three varieties of goal feedback, namely undirected (i.e., with goal input units set to zero after the initial step), self-directed (i.e., with goal input on each cycle set to the goal output from the previous cycle), and SS-directed (i.e., with goal input set to the correct values to guide behavior on a step-by-step basis). In order to establish mean
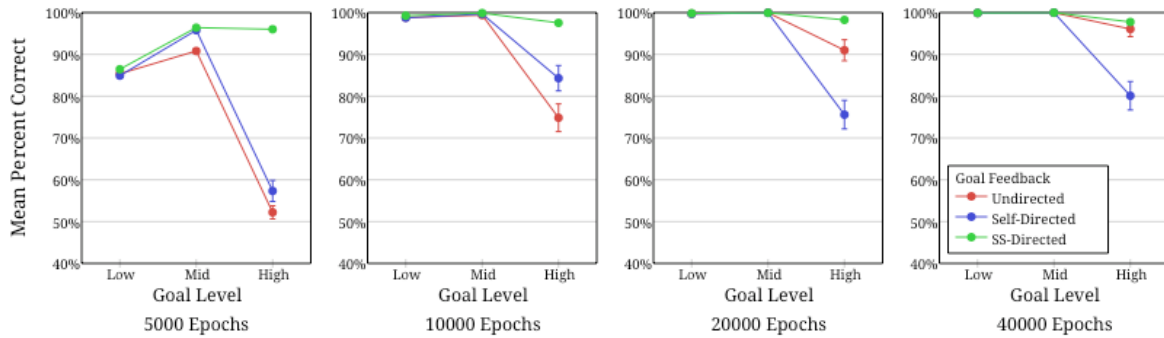
Figure 2: Mean percentage of correct trials for sequences corresponding to each goal level and with each form of feedback at four points in training. Error bars show standard error over 50 trained networks.

performance of the model across different values for initial weights this procedure was repeated for 50 different networks at each level of training.

Results

The graphs in Figure 2 show, for each level of training, the mean percentage of correct trials for each level of goal and with the three different forms of goal feedback. Consider first the case of undirected feedback (red line). After 5000 epochs of training the GC model with undirected feedback is able to generate correct sequences for low-level goals on approximately 85% of trials. Performance is slightly better on mid-level goals, but drops to 50% on high-level goals. With further training, performance with undirected goal feedback improves such that by 10,000 epochs the GC model performs near ceiling on low-level and middle-level goals and at 75% for high-level goals. With further training this increases to 95% by 40,000 epochs. Performance with self-directed goal feedback (blue line) follows a similar pattern, though initial learning is faster and later learning is slower. When goal inputs are set to reflect the current goals and subgoals on a step-by-step basis (SS-directed; green line), the GC model performs even better, such that after only 5,000 epochs of training, it is able to correctly generate sequences on 85% of trials for low-level goals and over 95% of trials on mid-level and high-level goals. After 10,000 epochs the model performs near ceiling for all goal levels. Although not shown in the figure, performance on high-level goals in the undirected and self-directed conditions continues to improve with training, such that after 100,000 epochs, models are correct on 99% of undirected trials and 92% self-directed trials.

Discussion

Simulation 1 demonstrates that the GC model is able to learn goal-directed sequences at all three levels, and moreover that it is able to generate such sequences either with goal input on just the first step (as in undirected mode), or with self-generated goal input on all steps (as in self-directed mode). The model is thus able to learn to use goal-input when it is present. This is demonstrated most clearly by the model's behavior in the SS-directed condition, when goal inputs are set to reflect the current goals and subgoals on a step-by-step basis. The guidance provided by the goal input units in this condition parallels that provided by supervisory control in deliberate or willed control of behavior, and these simulations demonstrate that once the model has acquired action sequences for low-level goals (after relatively little learning – 5000 epochs) it is possible to guide the model to perform complex action sequences corresponding to high-level goals.

The simulation also demonstrates how the model accounts for four of the eight characteristics of the control of sequential action highlighted in the introduction. First, that action may be

controlled at multiple levels (characteristic 4) is demonstrated by the performance of the model in SS-directed mode (Figure 2, green lines, all graphs) – with goal inputs set in this way the model performs near ceiling for low-, mid- and high-level goals. Second, that routine action requires minimal attentional control (characteristic 6) is shown by the near ceiling performance of the model for low- and mid-level goals in undirected mode after 10,000 epochs (when action sequences pertaining to the goals may be considered to be reutilized), and by the fact that for high-level goals performance approaches ceiling with further training (Figure 2, red lines, right-most graphs). Third, that learning reduces the need for attentional control (characteristic 7) is demonstrated by the increasing proportion of correct trials achieved in undirected mode with training (Figure 2, the red lines being higher on graphs towards the right than on graphs towards the left). Finally, that attentional control may initiate learned action sequences (characteristic 8) is demonstrated by the fully trained model's ability to correctly perform action sequences for all levels of goal without feedback (i.e., Figure 2, the red line in the right-most graph).

None of the above observations relate to the operation of the GC model in self-directed mode (i.e., Figure 2, blue lines). The difference between performance in self-directed and SS-directed modes can be viewed as a measure of the model's error – the error between the model's prediction of the next goal (which is fed back) and the actual next goal (as would be supplied by the supervisory system). For low-level and intermediate-level goals the GC model functions well in self-directed mode. It is only for high-level goals were self-directed mode is not effective. Closer examination of processing in these cases reveals that the model's performance is compromised by goal feedback that is insufficiently decisive. For example, in coffee making it is possible either to add sugar or cream after adding the grinds. The model therefore learns to associate both subgoals at strength 0.5 with completion of adding grinds, rather than either subgoal at strength 1.0.

Simulation 2: Intentional Control

One key motivation for the goal circuit is the desire to provide a mechanism whereby deliberate control can over-ride a learned action sequence. This is the scenario explored in Simulation 2. Specifically, consider the situation where one wishes to prepare a "non-routine" beverage, such as a cup of tea without sugar, or a cup of coffee with two sugars. Botvinick and Plaut (2004) considered precisely this scenario, and demonstrated that their original SRN could, with the addition of a further instruction unit, be trained to prepare coffee with either one spoonful or two spoonsful of sugar. Cooper and Shallice (2006) criticized the training required by the SRN for this modified task. Essentially the SRN had to be trained from scratch on the two variants of coffee-making – there was no transfer of learning from one task to the other and no possibility of instructing the original model, once it had learned to make coffee with one spoonful of sugar, to add a second spoonful. Cooper and Shallice instead argued that this variant of coffee making was likely to involve recruitment of the standard coffee-making routine, but with mechanisms for the support of non-routine behavior being deployed to repeat the "adding-sugar" subroutine at an appropriate point in time (i.e., mechanisms associated with the supervisory system). The GC model provides the means of operationalizing this proposal.

Method

The model was trained with the same parameter settings and goal specifications as in simulation 1. Five levels of training were considered (2,500, 5,000, 10,000, 20,000 and 40,000 epochs), and 50 networks were trained at each level, yielding a total of 250 trained networks.
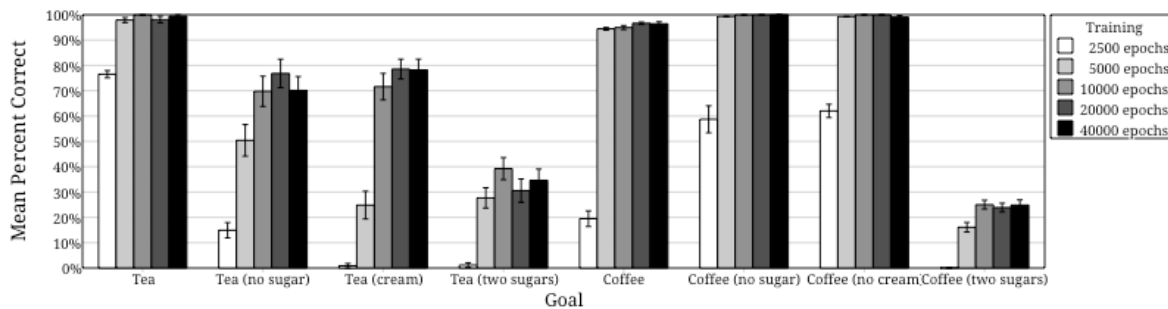
Figure 3: Mean percentage of correct trials for each goal level and with each form of feedback at four points in training with two levels of hidden unit noise (upper panel: noise s.d. = 0.10; lower panel: noise s.d. = 0.20). Note that in comparison to Figure 2 the range of the vertical axis has been increased.

Each trained network was tested on each of eight goals: the original tea-making and coffee-making goals from the training set, the goals of tea without sugar, tea with cream, and tea with two sugars, and the goals of coffee without sugar, coffee without cream, and coffee with two sugars. On each step during testing, goal input units were set to values that an assumed supervisory system would provide. For example, when the goal was to make tea without sugar, the goal units for the "drink" subgoal were activated immediately after the tea had been steeped (when during training steeping the tea was followed by "add sugar"). Each trained network was tested 100 times on each of the eight goals, yielding a mean accuracy for that level of training and that goal.

Note that in both training and testing, the effect of the *fixate-sugar* action was dependent upon the number of available sugar sources. In particular, if the sugar packet was used to add the first measure of sugar, then the packet would become empty and any subsequent attempt to *fixate-sugar* would result in fixation resting on the sugar bowl.

Results

Figure 3 shows the mean percentage of correct trials for each goal at each level of training. Unsurprisingly, the model performs almost perfectly at the routine "tea" goal and nearly as well at the routine "coffee" goal, even with relatively little training. This re-iterates the result shown in Figure 2 (green lines), of good performance under control from the supervisory system. The more interesting results arise from the novel goals. When instructed to prepare coffee without cream or without sugar, the GC model performs almost flawlessly, despite having never been trained to generate either sequence. The situation when making tea appears less satisfactory, where accuracy when skipping the sugaring subroutine or adding a creaming subroutine ranges from 70% to 80% after moderate training, but fails to increase with more extensive training. It is, however, sequences involving the addition of sugar twice that are most problematic. Here, while accuracy is greater for tea making than coffee making, in neither case does it reliably exceed 40%. Moreover, the situation again fails to improve with more extensive training. While levels of 40% to 80% accuracy are far from perfect, they are also well above the zero baseline one would expect if supervisory guidance was not possible.

Discussion

We begin with consideration of the major trends in the network's behavior, and in particular the slightly inferior performance on standard coffee making compared to standard tea making. The failure of the trained network to flawlessly generate an appropriate sequence for standard coffee making on each trial contrasts with the near perfect performance of the trained network

on standard tea making. Failures in coffee making are due to a tendency to occasionally omit the sugar or cream subroutines. This tendency is a result of the presence of the two variants of coffee making within the training regime – with cream and then sugar or with sugar and then cream. Even with substantial training, the network occasionally conflates these sequences, resulting in either sugaring or creaming being omitted. The complementary aspect of this is that it is relatively easy to guide the network when making coffee into deliberate omission of either subsequence, as when making coffee without sugar or without cream. That these are indeed complementary phenomena is demonstrated by the fact that the reverse pattern is shown with tea making, where during training there is no variation in the order of the subroutines. Here, we see almost flawless performance of the basic routine but relatively frequent errors when the network is guided to make tea without sugar or with cream.

When two sugars are required, two factors interact to limit the effectiveness of guidance from the supervisory system. First, recall that each sugaring subroutine begins with a fixate-sugar instruction, and this instruction may result in fixation moving to either the sugar bowl or (if it is not empty) the sugar packet. On some trials, both fixate-sugar instructions at the beginning of each sugaring sequence will result in fixation on the sugar bowl. In these situations the model as trained will fail, simply because the basic sugaring subroutine assumes, when using the sugar bowl, that the sugar bowl is closed. Yet that subroutine removes the sugar bowl's lid and does not replace it. The second attempt at sugaring will therefore fail, with an attempt to remove the lid from the open sugar bowl. The second factor depends on the overall task. If the task is tea making with two sugars, guidance from the supervisory system must overcome the learned subsequence transition that, at least when preparing tea, sugaring is always followed immediately by drinking. This subsequence is encoded in the internal weights of the hidden layer. Since it is highly routine in the context of tea making it is hard to over-ride through deliberate control. If the task is coffee making with two sugars, the situation is similar if the supervisory system chooses to attempt sugaring twice in succession (either before or after adding cream). Only if the supervisory system chooses to perform the subsequences in the order sugaring – creaming – sugaring will the transition probabilities between subsequences be consistent with the training set, and only then (and when different sources of sugar are chosen) is the network likely to be successful.[3]

These simulations therefore again demonstrate that the GC model satisfies characteristics 4 and 8 of the control of sequential action, namely that action may be controlled at multiple levels and that attentional control may initiate and/or override learned action sequences. The failure of the model, however, to perform perfectly on novel variations of the tea-making and coffee-making tasks may appear to be a cause for concern: phenomenologically, if one attends to a task one can direct action at will. There are at least two plausible responses to this concern. First, deliberate direction of the GC model is more successful when attempting to perform a novel variant of a goal-directed sequence that was itself variable during training. Extending the training set with more variants of tea-making or with other beverage preparation tasks which share subroutines of tea-making and coffee-making is therefore likely to result in a system that is more easily directed away from a single routine. (This is essentially an extension of an argument proposed by Botvinick & Plaut, 2006.) There is a trade-off, however, and such a network is likely to require substantially more training in order to acquire routines in the first place. Second, it may be that greater control from the deliberate route is required. This would seem to require an alternative architecture in which each action results from the network settling to a stable state (i.e., an attractor state), and where deliberate

---

[3] This has been confirmed with additional simulations, where this constraint, plus the training regime described in Simulation 3, led to correct performance of preparation of coffee with two sugars in over 95% of attempts.

control can influence that settling. The interactive activation network model of Cooper and Shallice (2000) is effectively a localist implementation of such an architecture.

Simulation 3: Environmental Variability

A significant limitation of the GC model as described thus far is that in order to complete any sequence – whether it be routine or novel – it is necessary for the initial state of the world to match the state assumed in the network's training regime. Thus, and as discussed above, while the model is trained to add sugar from either a packet or a closed sugar bowl, if it attempts to use the sugar bowl but the lid of the sugar bowl has already been removed (e.g., because it was not put in place by the previous user, or if the GC model is attempting to add sugar twice from the same bowl) then the model will fail in its attempt to add sugar. More generally, in real life behavioral flexibility demands that subroutines can be executed from a range of initial states. Consider the case of buttering toast. Normally one might begin this routine by fixating upon and picking up a butter knife, but if one is holding a knife (e.g., because one has just finished buttering another slice of toast), then these initial actions can and should be omitted. On the other hand if one is holding some other implement (e.g., a spoon) then it is necessary first to put down the spoon before fixating upon and picking up the butter knife. Simulation 3 demonstrates that with a generalized training regime the GC model is capable of this kind of behavioral flexibility.

Method

The model was trained with the same parameter settings and goal specifications as in Simulation 1. Again as in Simulation 1, 50 networks were trained in this manner for each of five levels of training, ranging from 2,500 epochs to 40,000 epochs. In contrast to earlier simulations, the initial state of the world was not fixed. Instead, on only 1 in 8 training trials was the network initially fixated on nothing. On the remaining trails it was initialized with fixation on one of the seven objects in the simulated world, each with equal probability. Similarly, on only 1 in 8 training trials was the network initially holding nothing. Again, on the remaining trials it was initialized to be holding one of the seven objects from the simulated world, each with equal probability. Finally, the four containers (coffee packet, sugar packet, sugar bowl, cream carton) were closed in 50% of trials and open in 50% of trials. All of these variable attributes of the initial state were set independently, meaning that there were 1024 possible initial states. Each epoch still involved one trial for each of the 18 training goals. Note though that on each trial (for both training and testing) the initial state was set randomly, so with 2,500 epochs it is likely that some of the 1024 possible initial states were never confronted, while even with 40,000 epochs, one would anticipate each initial state to occur only about 40 times during the entire training regime.

In order to evaluate the effect of this more general training on performance in general and guided novel behavior in particular, the network was tested as in Simulation 2. Thus, each trained network was tested with goal inputs set by an assumed supervisory system on tea-making, coffee-making, and each of the six novel goals. As in earlier simulations, each trained network was tested on each goal 100 times, yielding a mean accuracy for that level of training and that goal.

Results

Figure 4 shows the mean accuracy of the GC model under this revised training regime for each of the eight test goals. In comparison to Figure 3, accuracy is lower for the same level of training. This is to be expected given the variability in the initial state of the world. Note though that with sufficient training (40,000 epochs) performance for most goals is still very
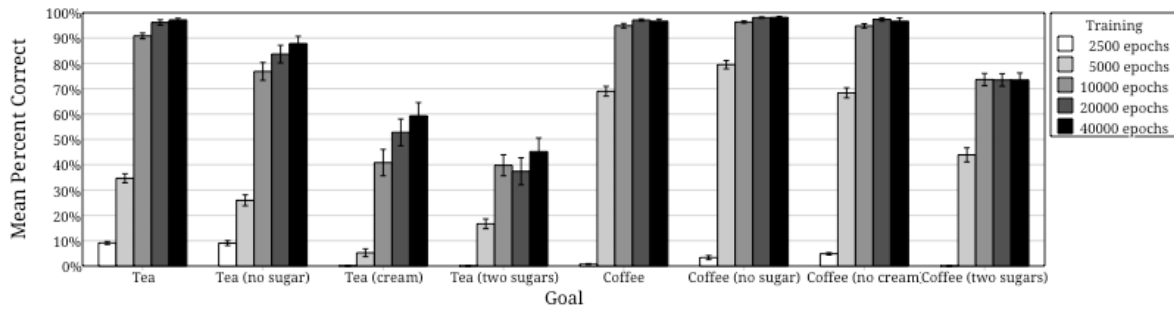
Figure 4: Mean percentage of correct trials for trained and novel goals at five points in training and with a variable initial state of the world (cf. Figure 4).

good. The key difference from Figure 3, however, is the greatly improved accuracy when preparing coffee with two sugars. Here, the model with supervisory-system guidance correctly performs the task on approximately 75% of trials. This contrasts with performance at the 25% level in Simulation 2.

Discussion

Great flexibility in action selection is required in order to correctly perform goal-directed action sequences in an uncertain or unpredictable world. For example, in order to achieve even a low-level goal of "getting" (i.e., picking up) a specified object, action selection must be sensitive to whether the object is already held, and if not, to whether the hands are free and to where fixation is directed. Thus, the goal may require anything between zero and three actions to achieve, depending on the state of the environment. Simulation 3 demonstrates that if trained appropriately the GC model can exhibit this flexibility, both at the lowest level of goals and at higher levels. In terms of the characteristics of the introduction, the simulation demonstrates that the GC model meets characteristic 3: that not all actions within a sequence are equal – the model can learn to include enabling actions when they are necessary but leave them out when they are not.

This characteristic is of course dependent upon having appropriate training – the model can only cope with variation in the environment if it has been exposed to similar variation during training. Note though that training on variant environments is not exhaustive. The training regime ensures that the model encounters a range of environments during the training phase, but it does not systematically ensure that each and every possible arrangement of objects is presented multiple times (or even once) for each and every goal.

General Discussion

We have argued that current psychological models of routine sequential action fail to provide a mechanistic account of the acquisition of flexible control, and presented the Goal Circuit model as a hybrid approach that resolves key difficulties related to this issue. We see the control of routine sequential action as an ability in the sense of Cassimatis et al. (2008), i.e., as a higher-order cognitive phenomenon exhibited by humans that must be explained. Thus, our approach has been to demonstrate how general and ubiquitous characteristics of routine sequential action emerge from the GC model. The model assumes two basic properties or characteristics of sequential action – that it is purposive and hierarchical – and exhibits six other characteristics enumerated in the introduction. It augments the model of Botvinick and Plaut (2004) in several important ways, most notably by providing the core SRN with an interface to the hypothesized SS. The interface enables routines at different levels to be accessed, in a fashion consistent with the interactive activation model of Cooper and Shallice

(2000), thus allowing for flexible control at the level of already acquired routines (e.g., online modifications, error correction and/or reordering of known subsequences). We hold this extension to be important because without it only behavior that was entirely routinized and evoked by external instruction or by object affordances could be captured. Our model thus implements the interplay between all *three* possible influences within the CS/SS complex: bottom-up triggering from the environment, ordering constraints from 'horizontal threads' and, depending on the control mode, additional top-down influences ('vertical threads') from the supervisory system. Importantly, the exact relation between these three possible influences is dependent on both experience and local task complexity.

Multiple routes and their interaction in the selection of action

As noted earlier, the GC model has three paths that combine to produce action selection. The direct path (cf. Figure 1a), mapping from perceptual input to action, may be seen as providing a baseline distribution of plausible actions given the current perceptual input but ignoring any contextual information. On its own, the direct pathway tends to produce a broad and non-specific distribution of possible actions to select at each instant, reflecting the fact that, as in real life, many actions are typically possible at any time, and situations in which one and only one action is physically possible are rare. The direct pathway allows some generalization – actions that are frequent in similar perceptual contexts will receive some excitation – but if an action has never been selected in the current perceptual context, or any similar perceptual context, then it will receive little or no excitation from this route. This tends to rule out an enormous number of possible successor actions (e.g., the physically impossible ones, such as attempting to tear open the sugar bowl, or trying to put something down when nothing is being held) and provides a useful bias towards possible and frequent actions in cases where the more sophisticated pathways fail (e.g., in error recovery or when the environment is different from expected, as has been shown with the open sugar bowl above in Simulation 3). To return to our analogy from the introduction: no matter how novel the path and how confused the rider, it will be hard to convince the horse to turn left when there is a tree standing in the way – possible paths are always restricted to physically valid choices. The direct pathway thus serves to suppress impossible actions and biases the basic level system towards the most likely action given the current perceptual input, reflecting frequency and, if training is assumed to be continuous, recency effects within the system's experience.

The recurrent inputs from the context units (cf. Figure 1b), in contrast, add a bias towards those successor actions that are plausible at this point when also taking implicit contextual information into account. In the GC model this implicit contextual information might comprise temporal context information in terms of prior states – what has been seen and done before – and also internalized goals – which goals have been active before. As both of these aspects are represented within the shared hidden layer, and previous hidden layer activation is used as implicit context, they both may have an influence on the selection of the current action. Adding this implicit context information will usually lead to a further restriction on the set of possible successor actions (see Figure 5). Actions that are physically possible, but inappropriate given this implicit context, can be ruled out. How correct and specific the recurrent contribution is depends on the salience of the implicit context information. It is this salience that increases with experience and as shown in simulation 1 (undirected mode) it is sufficient to guide performance in a moderately complex, but well learned task.

The salience of the implicit context information and thus the quality of the basic system's choice at each point in a sequence depends on several factors such as experience, distance of relevant dependencies, and neighborhood (i.e., number and frequency of locally similar
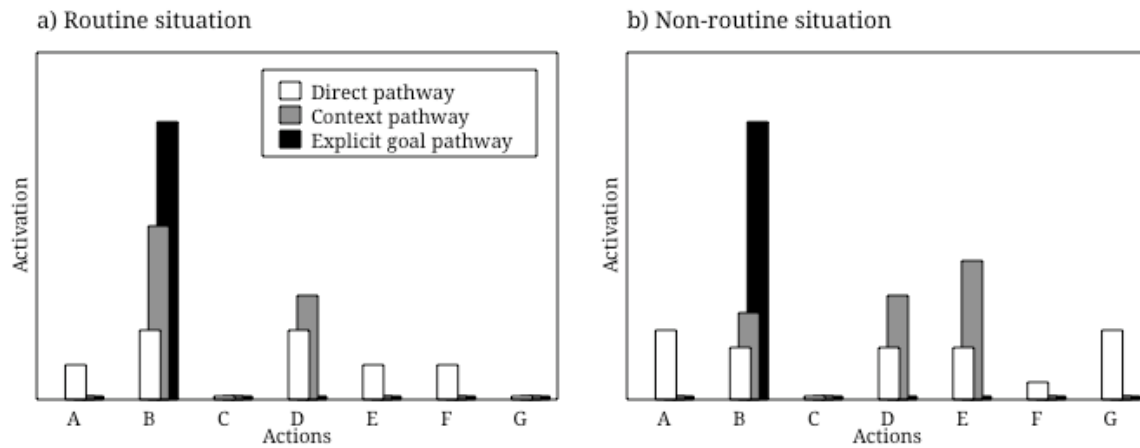
Figure 5: Activation that possible successor actions may receive from the different pathways in a hypothetical routine (Figure 6a, left) and non-routine (Figure 6b, right) situation. The activation from the direct pathway is shown in bars with a white fill, the contribution of the implicit context is depicted in gray, while the black bars stand for the bias from explicit goal input. In the routine case, all sources of activation support the same action (action B), while in the non-routine case multiple actions are supported by both the direct pathway and the context, but only one (again action B) is supported by the explicit goal input.

congruent or incongruent sequences). For those transitions in a sequence where the above-mentioned factors go in an unfavorable direction – novel, long, structurally complex sequences with many 'false friends' – the basic system (CS) might not be able to provide an unambiguous choice. It is in these cases that the goal circuit (cf. Figure 1c) can exert a decisive influence by further narrowing down the choice, and possibly disposing of the remaining ambiguity.

Figure 5 illustrates how the three influences combine. The general picture is one of a hierarchy of pathways that try to enforce an increasingly constrained set of transitions. Transitions advocated by higher-level pathways are a subset of the transitions that a lower level pathway proposes, because higher-level pathways take more information into account and thus produce less ambiguous choices of action. The difference in the distribution of transitions is most notable in cases where more information helps most (see Figure 5b), i.e., when temporal context has to be taken into account and/or when the information in the implicit context layer is not (yet) sufficiently salient. The choice implemented by the different pathways can be very similar, though, in situations where selection of the correct action is not much helped by additional context information or where the basic system has already internalized the explicit goal information (see Figure 6a). Whether explicit goals are activated or not does not make much of a difference in these latter (routine) situations – the basic (CS) system is sufficient or, as Wood and Neal (2007, p. 853) put it: "When habits and goals dictate the same response, […] goals in effect are rendered epiphenomena". Importantly, the relationship between the influences of the different pathways may change (a) for every step in a sequence and (b) through time as the experience of the network changes. The latter allows for a gradual transition of the responsibility for action selection from better informed pathways to less informed ones, where steps within invariant, tightly integrated subsequences become routinized first while the more informationally complex steps (such as branching points) may rely on the additional influence of SS for longer.

This proposed combination of influences contrasts with the uncertainty-based controller of Daw et al. (2005) that, they argue, selects between the outputs of the cached (i.e. habit-based) and tree-search (i.e. goal-based) systems. Yet the general architecture of the GC model mirrors ideas from research in both cognitive neuroscience and artificial intelligence. With regard to the former, Koechlin, Ody and Kouneiher (2003) have proposed that the neural architecture of cognitive control comprises a set of increasingly frontal processes that control behavior with respect "to stimuli, the present perceptual context, and the temporal episode in which stimuli occur" (Koechlin et al., 2003, p. 1181), with these processes being arranged along the lateral surface of the prefrontal cortex. With regard to the latter, architectures such as Brooks' subsumption architecture (see Brooks, 1991) conceive of cognition as a hierarchy of layers with increasingly complex functionality, with higher levels modulating lower levels. Glasspool (2005) argues that there is a close correspondence between the three levels emerging from convergent research in artificial intelligence, on the one hand, and from cognitive neuroscience on the other. The three different pathways in our model fit well with the levels generally distinguished in both disciplines.

Learning and progressive routinization

The existence of multiple routes combined with learning mechanisms within the GC architecture work together to produce redundant representations for the control of routine tasks. The function of goal units, which represent control information in the SS in localist terms, is to bias the lower level system towards task-appropriate actions. This bias is most critical at subtask boundaries when environmental cues provide poor predictors of subsequent actions. Through learning, this control information must become re-represented as a distributed representation within the model's context layer (i.e., within CS). While we have adopted standard back-propagation through time (BPTT) in the current implementation, we see learning as a bottom-up process, with sequences related to lower-level goals being acquired before those for higher-level goals. We refer to this process as "progressive routinization". We therefore have no strong commitment to the use of BPTT (or the tens of thousands of epochs of training that BPTT requires), and see it as an implementation detail in the sense of Cooper et al. (1996) – it is one of conceivably many learning algorithms that might show progressive routinization.

Progressive routinization, or assembling novel sequences from existing behavioral patterns and subsequently routinizing such novel behaviors, presupposes that the constituent behaviors and the supervisory system's ability to access them are already in place. At least at the beginning of one's life, however, this is unlikely to be the case. Thus, while progressive routinization might be the norm for the kind of everyday activities typically considered in empirical research on routine action, a bottom-up approach for the acquisition of at least the basic building blocks of behavior is still needed. Hierarchical reinforcement learning (see Botvinick, Niv & Barto, 2009) may provide a framework for combining these two aspects of learning. Hierarchical reinforcement learning is an augmented form of standard reinforcement learning (Sutton & Barto, 1998), which is itself an effective method of learning goal-directed behaviors and for which there is substantial neurophysiological support (e.g., Dickinson & Balleine, 2002; Daw et al., 2005). Hierarchical reinforcement learning combines exploration (of the space of possible actions) and exploitation (of acquired action/reward contingencies) with the kind of hierarchical structure that we have argued characterizes the action domain (characteristic 2 from above). In previous work with the IAN model we have shown that standard reinforcement learning can be used to acquire associations between the state of the environment and appropriate (single) actions (Cooper & Glasspool, 2001). We have also shown, within an SRN architecture similar to that of Botvinick and Plaut (2004), that so-

called temporal difference reinforcement learning can be used to acquire short goal-directed action sequences within a restricted domain (Ruh, Cooper & Mareschal, 2005). However, temporal difference reinforcement learning did not scale well to the acquisition of longer sequences in more complex domains. Hierarchical reinforcement learning appears to be one way of resolving this scaling problem.[4]

On the role of goal input units

The concept of a goal has a long and checkered history within psychology. It was unfashionable during the behaviorist era, and served as a key ingredient in the cognitive revolution (Miller, Galanter & Pribram, 1960). The concept remains central to many contemporary models of both routine sequential behavior and skill acquisition (e.g., Fu & Anderson, 2006; Taatgen et al., 2008; Trafton, et al., 2011). In sequential action goals serve two functions. First, it is an action sequence's goal which makes the actions within the sequence cohere and allows them to be described as a unit (as in the case of the *add sugar* goal). Second, goals allow action sequences to be assembled in constructive (i.e., generative) ways. Thus, the beverage sweetening sequence should only be included in a larger beverage preparation sequence if that beverage requires sweetening, but sweetening may also be achieved by other means (e.g., by using an artificial sweetener), or it may be modified if extra sweetener is required, or it may be suppressed from a routine if one is attempting to reduce the sugar in one's diet (as in Simulation 3; See Cooper & Shallice, 2006, and Duncan, 1986, for additional discussion).

The goal units, when active, bias the basic level network towards the transitions that lead to reaching this goal, thus enforcing a specific "policy" (i.e., a specific set of input-action associations). Goals at different levels are redundant when the policy of the lower level goal is included in the policy for the higher-level goal, e.g., as in the case of *add sugar*, which is included in *make coffee*. However, higher-level policies include a preference for what should be done after each lower level goal is achieved (e.g., if preparing coffee, then after sugar has been added cream should be added), while lower-level policies do not (e.g., if adding sugar to a beverage as a stand-alone procedure, then no specific actions should follow once adding sugar is complete).

As in other models where goals are explicitly represented (e.g., Taatgen et al., 2008; Trafton et al., 2011), goal units also give the GC model a form of compositionality, in the sense that an action sequence related to a subgoal (e.g., *add sugar*) can be invoked (and correctly performed) in different contexts by activating the appropriate goal units. While compositionality might appear at odds with the parallel distributed processing substrate of the model, Plaut and McClelland (1993) have demonstrated in a model of reading that, if the domain is appropriately structured, gradient descent learning algorithms such as back-propagation through time yield so-called "componential attractors" – internal representations that reflect the compositional structure of the domain. A key difference between training of the GC model and that of Botvinick and Plaut's (2004) SRN is that the GC training set posseses compositional structure, and while we are not committed to the precise details of the

---

[4] These reinforcement learning methods are all model-free methods, in that they rely only on positive or negative reinforcement at various points in time, with the network choosing on each step the action that has been reinforced most in the current context. We hold no strong commitment to model-free methods. Indeed, while action *selection* within the contention scheduling system is model-free, it may still be necessary to learn a model of the environment in order to predict feedback (see Dickinson & Balleine, 2002). In particular, an influential account of low-level motor control argues that errors are detected by a mismatch between predicted and actual feedback (Wolpert & Ghahramani, 2000), and there is no reason to suppose that similar systems would not operate at the contention scheduling level.

learning algorithm, this structure is reflected in its results.

An alternative way of looking at the goal units is as a "habit-goal interface". Wood and Neal (2007) identify several properties of the habit-goal interface, including: slowly accrued automaticity within the habit system, context sensitivity of habits, autonomy of strong habits, and several ways in which habits can influence goals and vice versa (e.g., goals can spur habit learning, habits can inform goals, and habits can interact with goals in guiding responses). Due to the flexible interplay of multiple routes the GC model addresses these properties in mechanistic terms.

With practice (or training) the need for goal input within the model is reduced (see Simulation 1). This is consistent with our own empirical findings. In a computerized version of a beverage preparation task, Ruh, Cooper and Mareschal (2010) demonstrated that participants were slower to select actions at subtask boundaries (so-called branching points) than at other points in the sequence, and that such selections were disproportionately affected when the task was performed in conjunction with an unrelated, attentionally demanding, secondary task. Importantly, however, interference with the secondary task was reduced with increasing practice. This was interpreted as a reflection of gradually decreasing demand for additional biasing from higher-level systems as the task became more well-learned or routinized (see also Ruh, 2007).

Taatgen et al. (2008) argue for a slightly different view of goal input during skilled behavior. They found in a complex cognitive task (setting and modifying a route within the Boeing 777 flight management system) that learning was faster, and supported better generalization, when instructions were given as sequences of preconditions / action pairs rather than as sequences of actions. The authors argue on the basis of this and related empirical and computational work for what they refer to as the minimal control principle, which requires that a "task representation should have a control structure that is as small as possible" (Taatgen et al., 2008, p. 550). Taatgen et al.'s concern in developing this principle was to maximize perceptual control without compromising action selection. While we concur on the utility of perceptual input, the GC model supports redundancy of representation, as described above and illustrated in Figure 5. A key consequence of this redundancy is that goal input can, at any time, over-ride acquired condition / action contingencies.

The Goal Circuit and the Supervisory System

We have attempted to capture how a hypothesized supervisory system could interface with a distributed contention scheduling system, rather than the detailed workings of the supervisory system itself. In testing the model we made use of simple ways to connect the predicted future goals to the goal input in the following step (the goal circuit), such as recirculating the predicted goals into the goal input in order to guide action selection at the next step, or simulating the functionality of the supervisory system by specifying the correct goal inputs at each step. Independently of the exact implementation of its functionality, however, the goal circuit facilitates the supervisory system's task in three ways:

1. It provides the supervisory system with a choice of possible goals to reach from the current state. This pre-selection rules out a great many other goals that the supervisory system does not have to worry about.

2. It gives an indication of whether it is necessary for the supervisory system to become involved (through output conflict or uncertainty). This is important because even in a

poorly learned task it is likely that there will be subsequences that are known from other contexts and that can be carried out without close supervision.

3. It functions through the activation of goal units, not individual actions, mirroring the biasing function of the supervisory system on the contention scheduling system within Norman and Shallice's (1986) Dual Systems framework.

This last point, that the goal circuit functions through activation of goals, supports flexibility in recovering from minor slips or in situations where there are many different ways to reach a specific goal (see also Taatgen et al., 2008). In addition it is consistent with empirical evidence and recent theoretical developments that attach central importance to the notion of goals as the representational format used by higher level executive systems for the purpose of overseeing, integrating and manipulating information from very dissimilar lower level functions (Braver & Cohen, 2000; Hommel, 2003; see Miller & Cohen, 2001, for an overview) and for managing the continuity of behavior over different time scales (Koechlin et al., 2003).

We have not attempted to model the detailed functioning of the supervisory system – the system that is assumed to be critical to the generation and regulation of all non-routine behavior. While some authors have suggested that the supervisory system is a homunculus with little explanatory power (e.g., Dennett, 1998), a series of papers by Shallice and colleagues (Shallice & Burgess, 1996; Shallice, 2002; Shallice et al., 2008; Shallice & Cooper, 2011) has attempted to fractionate the supervisory system into component processes such as goal generation, strategy generation, and monitoring and checking. These processes are hypothesized to work together in the control of complex cognition (such as when playing chess or planning a holiday), drawing on other non-central systems (e.g., language systems and episodic memory: see Shallice & Cooper, 2011) and interfacing with the contention scheduling system in the generation of behavior. However, with the possible exception of monitoring and checking, supervisory processes are not involved in routine sequential action, and it is for this reason that we have not attempted to model supervisory system processes here.

We conceive of the different control modes employed to test the GC model as a continuum that ranges from no contribution of the supervisory system to the supervisory system being fully involved in the current action selection. While fully routinized behavior involves a lengthy sequence of actions that may be activated appropriately in the absence of supervisory control, we suggest that the vast majority of naturalistic tasks are likely to involve a mixture of stretches of actions that are executed on 'autopilot', interspersed with occasional biasing input from the higher level control system. This view, coupled with the assumption that the supervisory system is a frontal system while contention scheduling draws upon parietal and sub-cortical structures (Shallice & Cooper, 2011), is also consistent with the neuropsychological evidence which suggests that impairments in routine behavior can follow both frontal and parietal lobe brain injury (Luria, 1966; Schwartz et al., 1991, 1998).

Conclusion

We have argued for eight simple characteristics of routine sequential action and presented a model of action control that captures these properties, either by design (characteristics 1 and 2) or as a consequence of that design (characteristics 3 to 8). The model goes beyond its predecessors by providing a computationally explicit account of the detailed interplay between all three influences identified in Norman and Shallice's (1986) dual-systems framework (bottom-up triggering, horizontal sequencing threads, and top-down control), thus

addressing the non-trivial problem of interfacing a basic habit or routine system with a higher level supervisory or goal system (see also Wood & Neal, 2007). It has been demonstrated that our extended SRN model is able to perform more or less routinized action sequences in a flexible, goal directed manner. While the model takes minor variations in the state of the environment in its stride, it also has the means to recover following error or to perform novel combinations of known subtasks by using its explicit goal representations. Moreover, the GC model is able to accommodate progressive hierarchical routinization through redundant encoding of (sub)task context within explicit goal and implicit context units. The model has the advantage over other existing models of acquiring this functionality in a psychologically plausible way (i.e., with shorter sequences first, with sequences associated with goals, and initially with tight control) and ending up with a functional interface through which an executive system may be used to guide the model's behavior by setting intentions or goals. Our model thus provides a computationally explicit approach that addresses not only how the rider (SS) may control the horse, but also how the horse (CS) can acquire the ability to carry out complex hierarchical routine tasks autonomously.

Acknowledgments

References

Altmann, E. M., & Trafton, G. J. (2002). Memory for goals: an activation-based model. *Cognition, 26*, 39-83.

Beldarrain, M. G., Grafman, J., Pascual-Leone, A., & Garcia-Monco, J. C. (1999). Procedural learning is impaired in patients with prefrontal lesions. *Neurology, 52*(9), 1853-1868.

Botvinick, M. M., & Bylsma, L. M. (2005). Distraction and action slips in an everyday task: Evidence for a dynamic representation of task context. *Psychonomic Bulletin & Review, 12*(6), 1011-1017.

Botvinick, M. M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review, 111*(2), 395-429.

Botvinick, M. M. & Plaut, D. C. (2006). Such stuff as habits are made on: A reply to Cooper and Shallice (2006). *Psychological Review, 113(4),* 917-927.

Botvinick, M. M., Niv, Y., & Barto, A. C. (2009). Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition, 113*, 262-280.

Braver, T. S. & Cohen, J. D. (2000). On the control of control: the role of dopamine in regulating prefrontal function and working memory. In S. Monsell and J. S. Driver (eds.), *Control of Cognitive Processes: Attention and Performance XVIII* (pp. 713-737). Cambridge, MA: MIT Press.

Brooks, R. A. (1991). How to build complete creatures rather than isolated cognitive simulators. In K. VanLehn (Ed.), *Architectures for Intelligence* (pp. 225-239). Hillsdale, NJ: Lawrence Erlbaum Associates.

Cassimatis, N. L., Bello, P., & Langley, P. (2008). Ability, breadth, and parsimony in computational models of higher-order cognition. *Cognitive Science, 32*, 1304-1322.

Cooper, R. P., Fox, J., Farringdon, J., & Shallice, T. (1996). A systematic methodology for cognitive modelling. *Artificial Intelligence, 85*, 3–44.

Cooper, R. P., & Glasspool, D. W. (2001). Learning action affordances and action schemas. In R.M French and J. Sougné (eds.) *Connectionist Models of Learning, Development and Evolution* (pp. 133–142). Springer-Verlag, London, UK.

Cooper, R. P., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology, 17*(4), 297-338.

Cooper, R. P., Schwartz, M.F., Shallice, T. & Yule, P. (2005). The simulation of action disorganisation in complex activities of daily living. *Cognitive Neuropsychology, 22*, 959–1004.

Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review, 113*, 887-916.

Daw, N. D., Niv, Y., & Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience, 8*, 1704-1711.

Della Sala, S., Marchetti, C., & Spinnler, H. (1991). Right-sided anarchic (alien) hand: A longitudinal study. *Neuropsychologia, 29*, 1113-1127.

Dennett, D. C. (1998). Reflections on language and mind. In P. Carruthers & J. Boucher (Eds.), *Language and Thought: Interdisciplinary Themes* (pp. 284-295). Cambridge University Press, Cambridge, UK.

Dickinson, A. (1985). Actions and habits: The development of behavioral autonomy. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 38*, 168-172.

Dickinson, A., & Balleine, B. (2002). The role of learning in the operation of motivational systems. In Paschler, H. & Gallistel, R., (Eds.) *Stevens' Handbook of Experimental Psychology, Third Edition, Vol.3: Learning, Motivation, and Emotion* (pp. 497-534). John Wiley & Sons: New York.

Duncan, J. (1986). Disorganisation of behavior after frontal lobe damage. *Cognitive Neuropsychology, 3*, 271-290.

Fu, W.-T., & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General, 135*, 184-206.

Gibson, J.J. (1977). The Theory of Affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving, Acting, and Knowing: Toward an Ecological Psychology* (pp. 67-82). Hillsdale, NJ: Lawrence Erlbaum.

Giovannetti, T., Schwartz, M.F. & Buxbaum, L.J. (2007). The coffee challenge: A new method for the study of everyday action errors. *Journal of Clinical and Experimental Neuropsychology, 29*(7), 690-705.

Glasspool, D. W. (2005). The integration and control of behavior: Insights from neuroscience and AI. In D. N. Davis (Ed.), *Visions of mind: Architectures for Cognition and Affect* (pp. 208-234). Hershey, PA: Idea Group Inc.

Goldberg, G., Mayer, N. H., & Toglia, J. U. (1981). Medial frontal cortex infarction and the alien hand sign. *Archives of Neurology, 38*, 683-686.

Heckhausen, H., & Beckmann, J. (1990). Intentional Action and Action Slips. *Psychological Review, 97*(1), 36-48.

Hommel, B. (2003). Planning and representing intentional action. *The Scientific World, 3*, 593-608.

Humphreys, G. W., & Forde, E. M. E. (1998). Disordered action schematas and action disorder syndrome. *Cognitive Neuropsychology, 15*, 771-811.

Humphreys, G. W., Forde, E. M. E., & Francis, D. (2000). The organization of sequential action. In S. Monsell & J. Driver (Eds.), *Control of Cognitive Processes, Attention and Performance XVIII* (pp. 427-442). Cambridge, MA: MIT Press.

Humphreys, G. & Riddoch, M. J. (2005). (Eds.). *Attention in Action*. Psychology Press.

James, W. (1890). *Principles of Psychology*. Henry Holt.

Jenkins, I. H., Brooks, D. J., Nixon, P. D., Frackowiak, R. S., & Passingham, R. E. (1994). Motor sequence learning: A study with positron emission tomography. *Journal of Neuroscience, 14*, 3775-3790.

Jueptner, M., Stephan, K. M., Frith, C. D., Brooks, D. J., Frackowiak, R. S. J., & Passingham, R. E. (1997). Anatomy of motor learning. 1. Frontal cortex and attention to action. *Journal of Neurophysiology, 77*, 1313-1324.

Koechlin, E., Ody, C. & Kouneiher, F. (2003). The architecture of cognitive control in the human prefrontal cortex. *Science, 301*, 1181-1185.

Land, M., Mennie, N. & Rusted, J. (1999). The roles of vision and eye movements in the control of activities of daily living. *Perception, 28*, 1311-1328.

Lhermitte, F. (1983). Utilisation behavior and its relation to lesions of the frontal lobes. *Brain, 106*, 237-255.

Luria, A. R. (1966). *Higher cortical functions in man*. New York: Basic Books.

MacKay, D. G. (1987). *The organisation of perception and action: A theory for language and other cognitive skills*. New York, NY: Springer Verlag.

Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual Reviews of Neuroscience, 24*, 167-202.

Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the structure of behavior*. New York: Henry Holt and Co.

Norman, D. A. (1981). Categorisation of action slips. *Psychological Review, 88*(1), 1-15.

Norman, D. A., & Shallice, T. (1986). Attention to action: Willed and automatic control of behavior. In R. Davidson, G. Schwarz & D. Shapiro (Eds.), *Consciousness and Self Regulation* (Vol. 4, pp. 1-18). New York, NY: Plenum.

Passingham R. E., Rowe, J.B. & Sakai, K. (2005): Prefrontal cortex and attention to action. In G. Humphreys & M. J. Riddoch (Eds.), *Attention in Action* (pp. 263-286): Psychology Press.

Plaut, D. C., & McClelland, J. L. (1993). Generalization with componential attractors: Word and nonword reading in an attractor network. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society* (pp. 824–829). Hillsdale, NJ: Erlbaum.

Reason, J. (1979). Actions not as planned: the price of automatization. In G. Underwood & R. Stevens (Eds.), *Aspects of Consciousness* (pp. 67-89). London, UK: Academic Press.

Reason, J. (1984). Lapses of Attention in Everyday Live. In K. Parasuraman & D. Davies (Eds.), *Varieties of Attention* (pp. 515-549). Academic Press.

Reason, J. (1990). *Human Error*. Cambridge, UK: Cambridge University Press.

Richer, F., Chouinard, M.-J., & Rouleau, I. (1999). Frontal lesions impair the attentional control of movements during motor learning. *Neuropsychologia, 37*, 1427-1435.

Riddoch, J. M., Edwards, M. G., Humphreys, G. W., West, R., & Heafield, T. (1998). Visual affordances direct action: Neuropsychological evidence from manual interference. *Cognitive Neuropsychology, 15*, 645-683.

Ruh, N. (2007). *Acquisition and control of sequential routine activities: Modelling and empirical studies*. PhD Thesis. Birkbeck, University of London, UK.

Ruh, N., Cooper, R. P., and Mareschal, D. (2005). A reinforcement model of sequential routine action. In Honkela, T., Köhönen, V., Pöllä, M., and Simula, O. (Eds.), *Proceedings of International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning* (pp. 65-70). Espoo, Finland.

Ruh, N., Cooper, R. P., & Mareschal, D. (2010). Action selection in complex routinized behaviors. *Journal of Experimental Psychology: Human Perception and Performance*, *36*(4), 955-975.

Schwartz, M. F., Montgomery, M., Buxbaum, L. J., Lee, S. S., Carew, T. G., Coslett, B. H., et al. (1998). Naturalistic action impairment in closed head injury. *Neuropsychology, 12*, 13-28.

Schwartz, M. F., Reed, E. F., Montgomery, M., Palmer, C., & Mayer, N. H. (1991). The quantitative description of action disorganisation after brain damage: A case study. *Cognitive Neuropsychology, 8*, 381-414.

Shallice, T. (1988). *From Neuropsychology to Mental Structure*. Cambridge University Press.

Shallice, T. (2002). Fractionation of the supervisory system. In Stuss, D.T. & Knight, R.T. (eds.) *Principles of Frontal Lobe Function* (pp. 261–277). Oxford University Press, Oxford, UK.

Shallice, T. (2006). Contrasting domains in the control of action: The routine and the non-routine. In Y. Munakata & M. H. Johnson (Eds.), *Processes of Change in Brain and Cognitive Development. Attention and Performance XXI* (pp. 3-29). Oxford: University Press.

Shallice, T., Burgess, N., Schon, F., & Baxter, D. M. (1989). The origins of utilization behavior. *Brain, 112*, 1587-1598.

Shallice, T., & Burgess, P.W. (1996). The domain of supervisory processes and temporal organisation of behaviour. *Philosophical Transactions of the Royal Society of London, B351,* 1405–1412.

Shallice, T. & Cooper, R. P. (2011). *The Organisation of Mind*. Oxford University Press, Oxford, UK.

Shallice, T., Stuss, D. T., Picton, T. W., Alexander, M. P., & Gillingham, S. (2008). Mapping task switching in frontal cortex through neuropsychological group studies. *Frontiers in Neuroscience, 2*, 79-85.

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

Taatgen, N. A., Huss, D., Dickison, D. & Anderson, J. R. (2008). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology: General, 137*(3), 548-565.

Trafton, J. G., Altmann, E. M., & Ratwani, R. J. (2011). A memory for goals model of sequence errors. *Cognitive Systems Research, 12*(2), 134-143.

Williams, R. J., & Zipser, D. (1995). Gradient-based learning algorithms for recurrent networks and their computational complexity. In Y. Chauvin & D. E. Rumelhart (Eds.), *Backpropagation: Theory, Architectures and Applications* (pp. 433-486). Hillsdale: Erlbaum.

Wolpert, D. M. & Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience, 3*, 1212-1217.

Wood, W., & Neal, D. T. (2007). A new look at habits and the habit-goal interface. *Psychological Review, 114*(4), 843-863.