Psychological Review
2004, Vol. 111, No. 2, 395–429

Doing Without Schema Hierarchies: A Recurrent Connectionist Approach to Normal and Impaired Routine Sequential Action

Matthew Botvinick
Department of Psychology, Carnegie Mellon University,
Center for the Neural Basis of Cognition, and
Western Psychiatric Institute and Clinic

David C. Plaut
Departments of Psychology and Computer Science,
Carnegie Mellon University,
and the Center for the Neural Basis of Cognition

In everyday tasks, selecting actions in the proper sequence requires a continuously updated representation of temporal context. Many existing models address this problem by positing a hierarchy of processing units, mirroring the roughly hierarchical structure of naturalistic tasks themselves. Although intuitive, such an approach has led to a number of difficulties, including a reliance on overly rigid sequencing mechanisms and a limited ability to address both learning and context sensitivity in behavior. We consider here an alternative framework in which the representation of temporal context depends on learned, recurrent connections within a network that maps from environmental inputs to actions. Applying this approach to the specific, and in many ways prototypical, everyday task of coffee-making, we examine its ability to account for several central characteristics of normal and impaired human performance. The model learns to deal flexibly with a complex set of sequencing constraints, encoding contextual information at multiple time-scales within a single, distributed internal representation. Mildly degrading this context representation leads to errors resembling the everyday "slips of action" that normal individuals commit under distraction. More severe degradation leads to a pattern of disorganization resembling that observed in *action disorganization syndrome*, a variety of apraxia. Analysis of the model's function yields novel, testable predictions relevant to both normal and apraxic performance. Taken together, the results indicate that recurrent connectionist models offer a useful framework for understanding routine sequential action.

Introduction

Much of everyday life is composed of routine activity. From the moment of getting up in the morning, daily living involves a collection of familiar, typically unproblematic action sequences such as dressing, eating breakfast, driving to work, etc. Because such activities can be executed without intense concentration, it is easy to overlook their psychological complexity. In fact, even the most routine everyday activities may call for a sophisticated coordination of perceptual and motor skills, semantic memory, working memory, and attentional control. Given the central role such activities play in naturalistic human behavior, it seems important to understand their psychological underpinnings.

In the present paper, we advance, in basic form, a theory of how routine object-oriented action sequences are performed. The framework we put forth, expressed in a set of computer simulations, takes as its point of departure existing work using recurrent connectionist networks. Applying such models to routine sequential action results in an account that differs sharply from most competing theories. As we shall discuss, most current accounts of action begin by assuming a processing system that is explicitly hierarchical in structure, and which contains processing elements that are linked, in a one-to-one fashion, with specific segments of behavior. The work we present here converges on two central theoretical claims that differentiate it from such hierarchical accounts: 1) The skills reflected in routine sequential activity cannot be identified with discrete, isolable knowledge structures, but instead emerge out of the interaction of many simple processing elements, each of which contributes to multiple behaviors; and 2) the detailed mechanisms that underlie routine action develop through learning, and as a result are closely tied to the structure of particular task domains. As we shall demonstrate, these tenets allow the present account to overcome some of the difficulties associated with theories based on processing hierarchies.

Hierarchical Organization in Routine Tasks

An essential point concerning the sequential structure of routine tasks was made early on by Lashley (1951). His aim was to point out the insufficiency of then current association-

Financial support for this research was provided by NIMH Grant MH16804 and a grant from the American Psychiatric Association and Eli Lilly and Company to the first author, and an NIH FIRST award to the second author (Grant MH55628). The computational simulations were run using LENS (Rohde, 1990), `http://www.cs.cmu.edu/~dr/Lens/`. We thank Marlene Behrmann, Laurel Buxbaum, Myrna Schwartz and the CMU PDP research group for helpful comments and discussion.

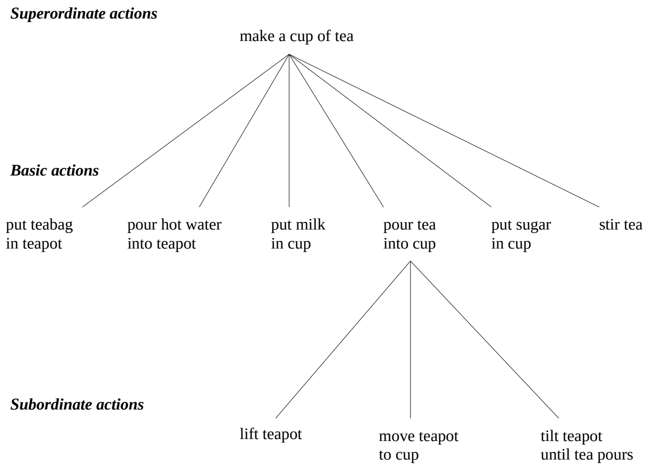

Figure 1. Hierarchical representation of a routine sequential task. Adapted from Humphreys and Forde (1999).

ist accounts of sequencing, which characterized serial behavior as a chain of simple links between each action and the next. Lashley noted that because individual actions can appear in a variety of contexts, any given action may be associated with more than one subsequent action. In such cases, information limited to the action just performed provides an ambiguous cue for action selection. Lashley's conclusion was that, in addition to representations of individual actions, the actor must also have access to a broader representation of temporal context, a "schema" that somehow encodes the overall structure of the intended action sequence.

Later work has extended Lashley's argument by emphasizing that sequential action typically has multiple, hierarchically organized levels of structure (e.g., Miller, Galanter, & Pribram, 1960; Schank & Abelson, 1977; Grafman, 1995). As stated by Fuster (1989, p. 159),

> Successive units with limited short-term goals make larger and longer units with longer-term objectives. These in turn, make up still larger and longer units, and so on. Thus we have a pyramidal hierarchy of structural units of increasing duration and complexity serving a corresponding hierarchy of purposes.

An illustration, drawn from recent work by Humphreys and Forde (1999), is shown in Figure 1. Here, simple actions occurring during a typical morning routine (e.g., lifting a teapot) are grouped together into subroutines (e.g., pouring tea into a cup), which are themselves part of larger routines (e.g., making tea), and so forth. Rather than being organized by a unitary schema, behavior here has been described as involving the coordination of multiple schemas, associated with different levels of temporal structure.

Hierarchical Models of Action

Since the original work highlighting the hierarchical structure of sequential behavior, a variety of proposals have been made concerning the cognitive mechanisms supporting such behavior. The most influential of these proposals share, as a central assumption, the idea that the hierarchical structure of behavior is mirrored in the gross architectural structure of the processing system. According to this approach, the processing system is arranged in layers corresponding to discrete levels of task structure, with processing at lower levels guided by input from higher ones. Models of this kind have proved partially successful in capturing basic phenomena relating to human sequential action. However, as we shall argue below, they also suffer from a set of important limitations.

One of the earliest attempts to model sequential action using a hierarchical processing architecture was by Estes (1972). This work posited a hierarchy of "control elements," which activate units at the level below. Ordering of the lower units depends on lateral inhibitory connections, running from elements intended to fire earlier in the sequence to later elements. After some period of activity, elements are understood to enter a refractory period, allowing the next element in the sequence to fire. This same basic scheme was later implemented in a computer simulation by Rumelhart and Norman (1982), with a focus on typing behavior.

Models proposed since this pioneering work have introduced a number of innovations. Norman and Shallice (1986) discussed how schema activation might be influenced by environmental events; MacKay (1985, 1987) introduced nodes serving to represent abstract sequencing constraints (see also Dell, Berger, & Svec, 1997); Grossberg (1986) and Houghton (1990) introduced methods for giving schema nodes an evolving internal state, and explored the consequences of allowing top-down connections to vary in weight; and Cooper and Shallice (2000) have employed "goal nodes" that gate activation flow between levels. Despite these developments, however, the majority of existing models continue to assume that the hierarchical structure of sequential behavior is directly reflected in the structure of the processing system, as a hierarchy of nodes or schemas.

An illustration of the state-of-the-art is provided by Cooper and Shallice (2000). Their model, illustrated in Figure 2, addresses the everyday routine of making a cup of coffee. As in earlier models, the processing system is structured as a hierarchy of nodes or units, with units at the lowest level representing simple actions, and nodes at higher levels representing progressively larger-scale aspects of the task.

Data Addressed by Hierarchical Models

The basic motivation behind the hierarchical approach is to address the issues originally raised by Lashley (1951), Miller et al. (1960), and related work. As such, the approach has been used to simulate normal behavior in a number of hierarchically structured task domains, including typing (Rumelhart & Norman, 1982), spelling (Houghton, 1990), and coffee making (Cooper & Shallice, 2000), among others.

Another appealing aspect of the hierarchical approach is

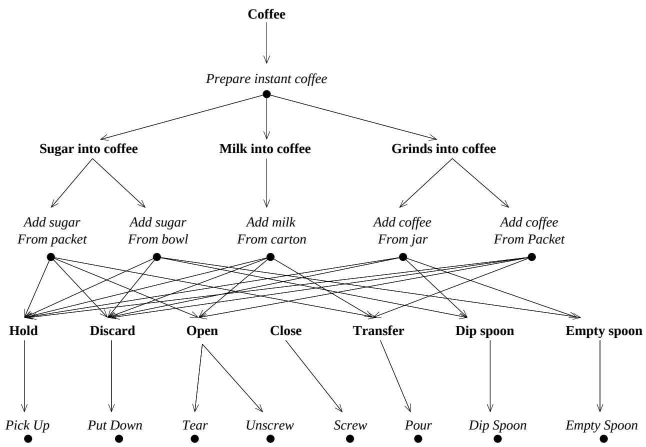

Figure 2. Processing architecture proposed by Cooper and Shallice (2000)

that it can also be used to provide an account of action pathologies. Specifically, hierarchical models have been applied to error data from two domains: everyday "slips of action" made by neurologically intact individuals (see, e.g., Baars, 1992; Cooper & Shallice, 2000; Norman, 1981; Reason, 1990; Roy, 1982; Rumelhart & Norman, 1982), and the behavior of patients with ideational apraxia and action disorganization syndrome (ADS), neuropsychological disorders involving impairments in performing sequential tasks using objects (described further below). In general, hierarchical accounts have led to two suggestions concerning the source of errors: disruptions of the within-level competition between schema nodes, and disruptions of the top-down influence of schema nodes on their children (for instances of both accounts, see Humphreys & Forde, 1999; MacKay, 1985, 1987; Norman, 1981; Rumelhart & Norman, 1982; Schwartz, Reed, Montgomery, Palmer, & Mayer, 1991). Once again, the most recent and detailed account is that of Cooper and Shallice (2000), who used interventions of both kinds to simulate various types of action slip and several key aspects of ADS.

Difficulties with the Hierarchical Approach

Despite its successes, the hierarchical approach has also encountered a number of persistent difficulties. In the following sections, we focus on four particularly basic problems.

Learning. A first problem with the hierarchical approach is that it has proved difficult to specify a learning mechanism by which schema hierarchies might develop. A central impediment in this regard derives from hierarchical models' reliance on localist representations, the use of single computational elements to represent supposedly discrete components of behavior. As illustrated above, such localist coding means that single units often code for composite structures. For example, a single computational element may be dedicated to representing the overall task of making coffee (see Figure 2). Since it is implausible to assume that a coffee-making schema is innate, hierarchical models are faced with the task of explaining how such a unit arises within the system. The discrete, localist nature of schema representation within such models has made it difficult to specify such an account. (We will consider one of the few existing attempts later in this article.)

Sequencing. A second problem with the hierarchical approach is the difficulty its proponents have encountered in specifying an adequate sequencing mechanism. The earliest approach (Estes, 1972; Rumelhart & Norman, 1982), which involved lateral inhibitory connections, was unable to cope with situations in which the same items appear in more than one order across sequences. In more recent work, several other sequencing mechanisms have been proposed, but basic difficulties remain. For example, in Cooper and Shallice (2000), the top-down flow of activation to each unit is gated until the appropriate preceding actions have been completed. However, because no mechanism is specified for this "symbolic gating," the model ends up assuming an important part of the functionality it is intended to explain. In a more explicitly mechanistic approach, Houghton (1990, see also Grossberg, 1986) has introduced compound units, with time-varying states, that can be used to activate lower-level units in sequence. Unfortunately, it is difficult to see how this approach might extend to tasks with more than two levels of structure (e.g., the coffee task treated by Cooper & Shallice, 2000). In addition, the approach appears unable to deal with situations involving cross-temporal contingency, where choosing the correct action depends on retaining specific information about earlier actions.

In Houghton's model and others, a critical mechanism in sequencing is *reflex* (or *rebound*) *inhibition*, by which units are automatically inhibited after firing. An inherent problem with this mechanism, acknowledged by Cooper and Shallice (2000), is that it can be applied only to units at the bottom level of the hierarchy. Units at higher levels must remain active until the entire segment of behavior they represent is completed. Immediate self-inhibition is thus inappropriate for these units, and other mechanisms must be specified. Unfortunately, in order to cope with the issue, Cooper and Shallice simply stipulated that units above the lowest level remain active until all relevant subgoals have been achieved. The actual mechanisms responsible for goal-monitoring and schema inhibition were not indicated.

Dealing with Quasi-Hierarchical Structure. A third problem with traditional models involves their limited ability to cope with situations in which performance of a routine should vary with context. Although a given schema may be associated with multiple higher-level schemas (e.g., "add sugar" may be associated with schemas for making coffee and making a cake), there is typically no mechanism that allows the details of the lower level schema to change depending on the higher-level schema that recruits it.[1] In effect, the

[1] Some theories allow higher-level schemas to pass arguments to

structure of hierarchical models enforces the assumption that tasks themselves are strictly hierarchical.

In order to see why this type of context-dependence presents a problem for hierarchical models, consider the following scenario: A waiter working in a diner serves three different regular customers each morning, one who prefers one scoop of sugar in his coffee, one who prefers two scoops, and one who prefers no sugar. Modeling the waiter's coffee-making repertoire in a hierarchical model would present a dilemma. Given that each customer expects a different amount of sugar, should the sugar-adding routine be represented using one unit or several? Clearly, using one unit is inappropriate, since this does not allow the amount of sugar added to vary according to the individual being served. The alternative strategy of using several separate sugar-adding units ignores the fact that the different versions of sugar-adding are all instances of a single routine, and thus share a great deal of structure. The same dilemma arises at the level of the coffee-making task as a whole. Should the model represent coffee-making as a unitary schema, or as a set of independent schemas, each relating to a different customer? Because behavior in naturalistic tasks is often not strictly hierarchical, issues such as these pose an important problem for traditional models of action.

Accounting for Error Data. As noted earlier, hierarchical models have been employed to account for errors, both those of normal subjects and those of patients with apraxia. One recent model, in particular (Cooper & Shallice, 2000), is impressive in the range of data for which it accounts. Through various manipulations, Cooper and Shallice elicited errors in many of the major categories described in empirical studies, comparing their data explicitly with observations of apraxic patients reported by Schwartz et al. (1991, 1995).

However, despite its strengths, this recent work raises concerns about the ultimate ability of hierarchical models to account for error data in a comprehensive and parsimonious way. Cooper and Shallice (2000) found it necessary to use several different manipulations to produce different kinds of action slips. For example, while errors of omission were produced by weakening of top-down influence within the schema hierarchy, repetition or perseveration errors were attributed instead to insufficient lateral inhibition. Of perhaps greater concern is that the model failed to capture at least two important features of the empirical data concerning errors. First, without the addition of special mechanisms, the model did not produce errors involving the repetition of an entire subtask after one or more intervening subtasks—so-called recurrent perseveration (Sandson & Albert, 1984). Second, the model did not reproduce an important relationship between error rate and error content in ADS: that, across patients, as overall error rate increases, omission errors form a progressively higher proportion of all errors (Schwartz et al., 1998; also see Figure 17 and further discussion below).

One further behavioral phenomenon that presents difficulty for hierarchical models is the fact that, when slips of action occur, it is often apparent how the preceding actions might have led the actor into the wrong program of action. Norman (1981) refers to such situations as instances of "capture," and provides the following example: "[Soon after having played a game of cards], I was using a copy machine, and I was counting the pages. I found myself counting '1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King' " (p. 8). Another, well known example comes from William James (1890, p. 115), who describes how "very absent-minded persons in going to their bedroom to dress for dinner have been known to take off one garment after the other and finally to get in bed." According to James, the bedtime sequence intrudes "because that was the habitual issue of the first few movements when performed at a later hour." Because the representation of context in hierarchical models does not depend directly on which actions have been recently performed, it is unclear how such models might account for this aspect of errors.

An Alternative Approach

An alternative framework for understanding sequential behavior is offered by existing work on *recurrent connectionist networks*. Beginning with the pioneering work of Jordan (1986b) and Elman (1990, 1991, 1993), numerous studies have demonstrated the ability of such networks to produce sequential behavior resembling that of humans in a variety of domains, including spoken word comprehension and production (Christiansen, Allen, & Seidenberg, 1998; Cottrell & Plunkett, 1995; Dell, Juliano, & Govindjee, 1993; Gaskell, Hare, & Marslen-Wilson, 1995; Plaut & Kello, 1999), lexical semantics (Moss, Hare, Day, & Tyler, 1994), reading (Pacton, Perruchet, Fayol, & Cleeremans, 2001; Plaut, 1999), sentence processing (Allen & Seidenberg, 1999; Christiansen & Chater, 1999; Rohde, 2002; Rohde & Plaut, 1999), implicit learning (Cleeremans, 1993; Cleeremans & McClelland, 1991), dynamic decision making (Gibson, Fichman, & Plaut, 1997), motor control (Jordan, Flash, & Arnon, 1994), and cognitive development (Munakata, McClelland, & Siegler, 1997). In the work to be reported here, we adapt the recurrent connectionist framework to the domain of routine, object- and goal-oriented sequential behavior, evaluating its ability to address a fundamental set of empirical phenomena.

The account we will put forth differs most strikingly from the hierarchical approach in the way that it portrays the representation of sequence knowledge. Rather than attempting to make such knowledge explicit, by linking it to specific elements within the processing system, the present account suggests that knowledge about sequential structure inheres in the emergent dynamical properties of the processing system as a whole. In the framework we will put forth, there is no isolable structure that can be identified with a schema. Borrowing the words of Rumelhart, Smolensky, McClelland, and Hinton (1986, p. 20),

---

lower level schemas, allowing a degree of context-sensitive behavior. We point out some limitations of this "slot-filling" approach in the General Discussion.

> Schemata are not "things." There is no representational object which is a schema. Rather, schemata emerge at the moment they are needed from the interaction of large numbers of much simpler elements all working in concert with one another.

The knowledge that structures this interaction, on our account, is not represented locally as in hierarchical models. Instead, knowledge about a variety of action sequences is distributed and superimposed over a large set of connection weights among processing units. The result is a system that displays behavior that can be hierarchically structured but also flexible and context-sensitive. Furthermore, as we will show, the same properties that give such a processing system its power also make it susceptible to errors resembling those occurring in human performance.

Recurrent Networks: The General Framework

Connectionist or parallel distributed processing models (Rumelhart & McClelland, 1986) comprise a set of simple processing units, each carrying a scalar activation value. The activation of each unit is based on excitation and inhibition received from units linked to it through weighted synapse-like connections. Often, the units in connectionist networks are segregated into three populations or 'layers.' A first layer carries a pattern of activation representing some input to the system. Activation propagates from this layer through an internal or hidden layer, which transforms the input information, sending a pattern of activation to an output layer whose units together represent the system's response to the input.

A network is described as *recurrent* when loops or circuits can be traced through its set of connections. For example, in the so-called simple recurrent network architecture, each hidden unit is connected to every other. A critical aspect of such recurrent connectivity is that it allows information to be preserved and transformed across time. On each step of processing, the network's recurrent connections carry information about the state of the system on the previous time step. Because this state carries information about earlier events, it allows the network to act in a way that is sensitive to temporal context.

The ability of recurrent networks to map from inputs to appropriate outputs and to encode, preserve, and utilize information about temporal context depends on the pattern of connection weights among its units. Using a connectionist learning procedure such as back-propagation (Rumelhart, Hinton, & Williams, 1986), an effective set of weights can be learned through repeated exposure to correct sequential behavior. Through the gradual, adaptive adjustment of its connection weights, the system learns to produce internal representations—patterns of activation across its hidden layer—that both facilitate the immediate selection of outputs and preserve information that will be needed later in the sequence.

The basic properties of recurrent connectionist networks, as demonstrated in prevous work, suggest that they may offer an appealing alternative to hierarchical models in the domain of routine sequential action. Unlike hierarchical models, recurrent networks are intrinsically suited to sequential domains, containing an inherent mechanism for structuring behavior in time. Because recurrent networks can learn, there is no need to stipulate task-specific structure at the outset. Rather than having the necessary representations built in, recurrent networks learn to represent current and past events in a fashion suited to task demands (Elman, 1990; McClelland, St. John, & Taraban, 1989). Importantly, recurrent networks are capable of encoding temporal structure at multiple time-scales simultaneously (Cleeremans, 1993; Elman, 1991), pointing to a capacity to cope with hierarchically organized sequential structure. At the same time, as Elman (1991) has shown in models of sentence processing, recurrent networks do not suffer from the context insensitivity of hierarchical models, having the capacity to integrate information across multiple levels of temporal structure (see also Servan-Schreiber, Cleeremans, & McClelland, 1991).

Despite these appealing aspects of recurrent connectionist networks, widespread skepticism toward such models appears to exist among researchers studying routine sequential action. For example, Houghton and Hartley (1995) have suggested that recurrent networks necessarily suffer from the same limitations as "chaining" models (see also Henson, Norris, Page, & Baddeley, 1996; Vousden, Brown, & Harley, 2000). Brown, Preece, and Hulme (2000, p. 133) argue that recurrent networks lack "temporal competence... the intrinsic dynamics that would enable them to progress autonomously through a sequence." Others have expressed specific doubt concerning the ability of recurrent networks to account for error data. For example, Cooper and Shallice (2000, p. 329) write,

> We know of no work in which such networks have been shown to be able to account for errors of the type observed in the action domain (specifically omission and other sequence errors). The principal difficulty in obtaining such errors within recurrent networks appears to arise from the lack of any separate representation of hierarchical relations (i.e., source/component schema relationships) and order information (i.e., the relative ordering of component schemas within a single source schema). It is thus difficult for order information to be disrupted without disruption to hierarchical relations.

Our work will attempt to demonstrate that such skepticism is misplaced.

Modeling Naturalistic Action

The goal of applying the recurrent connectionist framework to routine sequential action raises several implementational issues. First, since everyday action typically involves action on objects, it is necessary to formulate a way to represent not only actions but also their targets and implements. Second, since actions often alter the perceived environment, it is necessary to allow this to occur in the model. Third, in approaching error data, it is necessary to motivate a technique for inducing dysfunction. In what follows, we detail our approach to these issues.

Action on Objects. Allport (1987, p. 395) has noted, "Systems that couple 'perception' to 'action' must deal moment by moment with two essential forms of selection: Which action? and Which object to act upon?" Because computational models of action have often dealt with tasks that do not involve direct physical action on objects (e.g., language tasks), they have typically focused only on the first of these two forms of selection. Thus, a central question facing models of routine naturalistic action is how objects are identified as targets for action.

One promising hypothesis in this regard is that targets for action are specified *indexically*. That is, actions are directed toward whatever object is currently at the system's focus of orientation, where orientation can mean the point of visual fixation or, more generally, the focus of attention. This strategy, otherwise known as a "deictic" (Agre & Chapman, 1987; Ballard, Hayhoe, Pook, & Rao, 1997) or "do-it-where-I'm-looking" (Ballard, Hayhoe, Li, & Whitehead, 1992) strategy, has seen wide application in engineering and robotics (Whitehead & Ballard, 1990; McCallum, 1996). More importantly, it has been proposed as a model for how objects are selected as targets for action in human behavior (Agre & Chapman, 1987; Ballard et al., 1997, see also Kosslyn, 1994; Pylyshyn, 1989; Ullman, 1984).

The three-layer recurrent network architecture described earlier lends itself naturally to the use of indexical representation. One need only assume that the input layer, now interpreted as carrying a representation of the perceived environment, conveys information about which object is currently the focus of attention. Units selected in the model's output layer, now understood as representing actions, can be interpreted as directed toward that object. One potential implementation of this approach is diagrammed in Figure 3. Here, the input layer contains a segment labeled "fixated object," which specifies the visual features of the object currently at the focus of visual attention. The units in the output layer correspond to actions to be directed toward this object.

Some actions involve objects not only as targets but also as instruments or tools. Again following previous deictic models (e.g., Ballard et al., 1992), we assume that this role is assigned to whatever object the agent currently has in hand. Accordingly, the input layer in Figure 3 includes a second portion labeled "held object," which specifies the features of this object. Just as the fixated object is interpreted as the target for action, the held object (if any) is interpreted as the implement to be used.

Because, within this framework, actions are directed at whatever object is currently the focus of attention, selecting a new target for action necessarily involves shifting that focus to a different object. To this end, computational models using indexical representations typically involve not only *manipulative* actions (actions that involve transformation of the environment), but also *perceptual* actions, which serve to reorient to the system toward a new object (see Whitehead & Ballard, 1990). This can be understood as either a physical reorientation, such as an ocular saccade, or a covert change of focus accomplished through attentional adjustments. Units representing such perceptual actions can be incorporated into the output layer of the architecture diagrammed in Figure 3, with each unit representing an action such as "fixate the spoon."

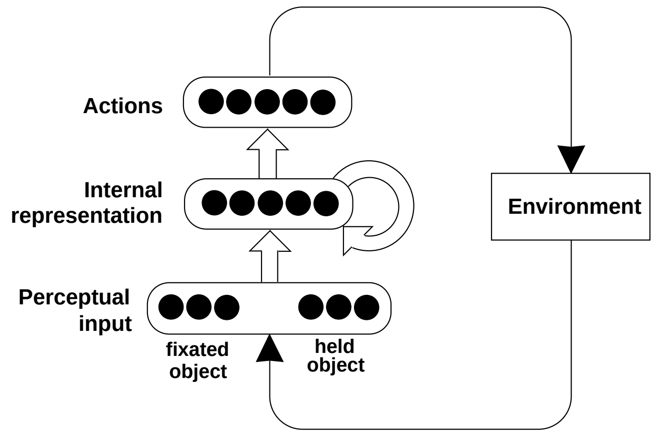

Figure 3. Architecture of the overall model. White arrows indicate that every unit in the sending layer is connected to every unit in the receiving layer. See text for details, including the number of units included in each layer.

Given this framework, sequential action on objects takes the form of a rough alternation between perceptual actions, which orient the system toward a target object, and manipulative actions, where the object is acted upon. Evidence for such an alternation in human behavior is provided by Ballard et al. (1992). They monitored hand and eye movements in a block-moving task, finding that subjects alternated between perceptual and manipulative actions, saccading to a location in the workspace, acting at that location, saccading toward a new location, and so forth. A similar alternation between perceptual and manipulative actions has been observed in naturalistic tasks. For example, Hayhoe (2000) monitored eye and hand movements as subjects prepared a peanut butter and jelly sandwich. A portion of the pattern observed is diagrammed in Figure 4. As in the task used by Ballard et al. (1992), performance here involved an alternation between visual orientation toward a target object and manipulative actions directed toward that object (see also Land, Mennie, & Rusted, 1998).

Implementing the Perception-Action Loop. An important aspect of naturalistic sequential action is that each movement, by altering the environment, can impact the perceptual input the system receives next. In order to capture this in a model, a functional representation of the environment must be interposed between the model's outputs and its subsequent inputs. The implementation diagrammed in Figure 3 incorporates such a simulated workspace. This maintains a repre-

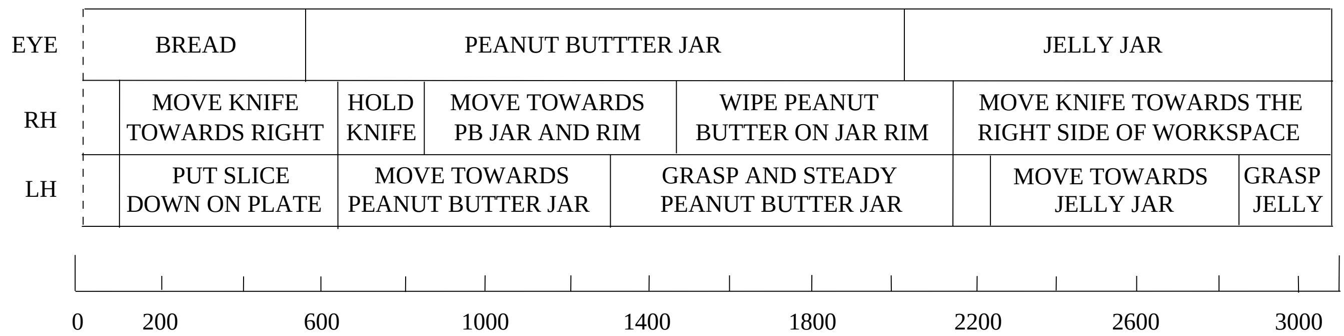

Figure 4. The pattern of visual fixations and manual actions observed by Hayhoe (2000) as subjects prepared a sandwich.

sentation of the state of various objects in the environment, updates this in response to each action, and, if appropriate, yields a new input pattern to the layers representing the objects currently fixated and held.

While some previous recurrent network models of sequencing have allowed model output to influence subsequent input, in general this has involved simply copying the model's output onto an input layer (e.g., Jordan, 1986b; although see Plaut & Kello, 1999). Interposing a simulated environment, as in Figure 3, permits us to address several important aspects of environmental feedback that are not encountered in that simpler case. First, in naturalistic behavior, elements of the environment can serve as a form of external memory (Agre, 1988; Rumelhart et al., 1986), providing cues about what actions have been completed. Importantly, the converse is also true: Significant changes to the state of the environment may not be reflected in perceptual feedback. In these cases, the system must rely on internal representations, rather than external input, if it is to behave appropriately (for a discussion, see Whitehead & Lin, 1995). A final point about environmental feedback is that it is, in general, probabilistic. Because there are many situations where a particular action can yield multiple results, the action system must be ready to deal with environmental contingencies that are only partially predictable.

Modeling Task Acquisition. The focus of the present research is on routine behavior. As such, we are more concerned with the outcome of learning than with the learning process itself. Nevertheless, a central claim of the present account is that experience plays a critical role in shaping the representations and mechanisms that support sequential behavior. Thus, the issue of learning provides an important part of the background for the account.

In human behavior, the acquisition of sequential routines can occur by a variety of means: explicit instruction, trial and error, problem solving methods, etc. Two methods that appear to be particularly important in everyday life are learning through prediction, and learning with scaffolding. As characterized by Schank (1982), much of our knowledge about action sequences is gained through a process of continual prediction-making; learning occurs when our predictions about actions and events turn out to be erroneous. One instance of such prediction-based learning would be learning through observation, where the learner follows the performance of an individual already familiar with the task, and attempts to predict his or her actions at every step. Scaffolding involves a similar process, except that the learner attempts to perform the task, with a teacher intervening only when the learner falters (Greenfield, 1984).

In both observational learning and learning with scaffolding, task acquisition is an active process. The learner attempts, on each step of performance, to produce (or predict) the next correct action, and learning occurs when this turns out to be incorrect. Our approach to simulating learning implements this basic process. In the simulations we will present, learning entails the step-by-step presentation of specific action sequences. On each step in the process, the network generates the representation of a possible next action, and learning occurs to the extent that this action fails to match the observed sequence.

Human learning of complex procedures appears to involve two principal stages, each depending on a functionally and anatomically distinct set of learning mechanisms: an initial phase, in which task knowledge is rapidly but superficially acquired, followed by a longer phase of consolidation or proceduralization (Anderson, 1987; McClelland, McNaughton, & O'Reilly, 1995; Schneider & Detweiler, 1988). The simulations we will present relate most clearly to the second of these phases, because they involve the establishment of highly routinized behavior through a very gradual learning process. The simulations implement no mechanisms for rapid binding. It is thus important to note that such mechanisms do not appear to be absolutely necessary for the acquisition of procedural knowledge; when they are bypassed through task manipulations (Stadler & Frensch, 1998), or impaired due to brain injury (Cleeremans, 1993; Cohen, 1984), a gradual form of sequence learning is still observed. Our simulations can be thought of as modeling this direct form of procedural learning. However, an alternative way of viewing the simulations is as modeling the process of consolidation. According to McClelland et al. (1995), consolidation occurs through a process by which long-term memory systems (housed in neocortex and basal ganglia) are trained by shorter-term (medial temporal lobe) learning mechanisms. The simulations can be interpreted as modeling this process, if the input and feedback provided to the model is viewed as coming not from the environment, but from a second learning system.

Modeling Dysfunction. Previous studies of action errors, in both normal subjects and individuals with apraxia, have regarded such errors as reflecting dysfunction in the basic mechanisms that give behavior its temporal structure. In hierarchical models, as we have discussed, this has involved disrupting either within-layer competition or the top-down influence of high-level schemas. In the present framework, the most direct way to compromise the mechanisms that support sequencing is to disrupt the information carried by the recurrent connections within the hidden layer. Several different methods can be used to induce such a disruption. Previous studies using recurrent networks have added random perturbations to connection weights (Dell et al., 1993) or to net inputs (Cleeremans, 1993), or reduced the gain of the unit activation function (Cohen & Servan-Schreiber, 1992). In the modeling work to be reported here, we added random noise to the activation values being conducted over the processing system's recurrent connections.

Importantly, the disruption of internal representations in the present framework can be viewed as corresponding, in terms of its consequences, to basic etiological factors underlying both slips of action and ADS. Studies of slips have emphasized that such errors tend to occur during periods of distraction, where there is cross-talk from task-irrelevant cognitive activity (Reason, 1990). We assume that internal representations of temporal context are among the representations affected by such cross-talk. The addition of noise to these representations can thus be understood as a functional correlate of mental distraction. More severe levels of noise can be interpreted as representing the effects of direct neural damage in ADS. While the basic problem here is structural rather than functional (as in the case of slips), we assume that at the computational level the two domains involve the same basic problem: a corruption of the system's internal representation of context.[2] Interestingly, there is independent motivation for using a single technique for modeling both slips of action and errors in apraxia. Based on observations concerning the patterns of errors made by normals and patients with apraxia of varying severity, Schwartz et al. (1998) have suggested that apraxia may represent an exaggeration of the same processes that lead to errors in normals (see also Roy, 1982).

Simulations

Using the approach just outlined, we conducted a series of computer simulations evaluating the capacity of a recurrent network to account for a variety of basic phenomena pertaining to routine sequential action. The simulations centered on a single model, trained on a set of concrete, everyday tasks—most centrally the task of making a cup of instant coffee. The behavior of this model was first compared with normal, error-free human performance (Simulation 1). Then, by impairing the mechanisms responsible for maintaining contextual information, the model was used to address slips of action (Simulation 2) and the behavior of patients with ADS (Simulation 3). We also carried out two additional simulations, in which the model was trained on other training sets, in order to address the specific issues of context sensitivity (Simulation 1a) and the effect of varying task frequency (Simulation 2a).

Simulation 1: Normal Performance

In the initial simulation, we asked whether the recurrent network framework could be used to account for a set of core features of error-free routine sequential activity. The relevant phenomena—several of which we have already mentioned—were gleaned from existing work in the domain, and include the aspects of action previously addressed by hierarchical models:

1. Routine sequences tend to assume a roughly hierarchical form (Miller et al., 1960; Schank & Abelson, 1977).
2. Elements at any level of this hierarchy may appear in multiple contexts (Lashley, 1951).
3. Although the environment often provides cues to the correct action, the information it conveys is also frequently insufficient to guide action selection without some added context (Whitehead & Lin, 1995).
4. In some cases, it may be permissible to execute the elements of a sequence in variable order.
5. In some cases, actions or subroutines may be substituted for one another.
6. The details of certain sequences may depend on the context in which they are performed (see Simulation 1a).

As an exemplar, the task of coffee-making is appealing for several reasons. To begin with, coffee-making involves all of benchmark features of routine sequential behavior enumerated above. Perhaps because of this, coffee-making, and its close relative tea-making, have figured in numerous empirical studies of routine behavior, including studies of slips of action (Reason, 1990; Humphreys, Forde, & Francis, 2000), and action disorganization syndrome (Schwartz et al., 1991, 1998; and see Lehmkuhl & Poeck, 1981). Furthermore, coffee-making has served as the focus for a recently proposed hierarchical model of sequential action (Cooper & Shallice, 2000). Addressing this task thus facilitates a comparison between approaches.

Methods

Task and Representations. Our formalization of the coffee-making task is presented in Table 1. (As discussed below, the sequence shown corresponds to one of four versions of the task that were use in training.) The task consists of a series of discrete steps, each involving a set of perceptual

[2] At a more concrete level, the addition of noise may be interpreted as an analogue for the effects of damage to long-fiber pathways due to head injury (a condition often associated with ADS), insofar as such damage is likely to reduce the fidelity of information-transmission across these pathways.

Table 2
Object Features and Actions

Fixated input	Held input	Action
cup	cup	pick-up
1-handle	1-handle	put-down
2-handles	2-handles	pour
lid	lid	peel-open
clear-liquid	clear-liquid	tear-open
light	light	pull-open
brown-liquid	brown-liquid	pull-off
carton	carton	scoop
open	open	sip
closed	closed	stir
packet	packet	dip
foil	foil	say-done
paper	paper	fixate-cup
torn	torn	fixate-teabag
untorn	untorn	fixate-coffee-pack
spoon	spoon	fixate-spoon
teabag	teabag	fixate-carton
sugar	sugar	fixate-sugar
instruct-coffee	nothing	fixate-sugarbowl
instruct-tea		

inputs and an associated action. As introduced above, perceptual inputs are divided into those pertaining to the object currently viewed and those pertaining to the object currently grasped. In both cases, objects are represented as a collection of complex, but in all cases perceptible, features. The full feature set employed is listed in Table 2.

Also as shown in Table 2, actions were represented by single descriptors.[3] Actions fell into two broad categories: manipulative actions that alter the environment (e.g., pick-up, pour, tear) and perceptual actions that orient the system toward a new object (e.g., fixate-cup, fixate-spoon). In keeping with the indexical representational scheme described above, the designations of manipulative actions did not specify which object was being acted on.

The sequence shown in Table 1 was one of four instances of the coffee-making sequence included in the training set. Each version of the task included four basic subtasks: 1) add coffee grounds to the hot water, 2) add cream, 3) add sugar, and 4) drink. However, the order of these subtasks varied. Specifically, in two sequences sugar was added before cream, and in the other two, cream before sugar. In addition, the training set also contained one subtask (sugar-adding) that appeared in two different forms. In two of the target sequences, sugar was added from a sugar bowl, in the other two, from a packet. Crossing these two dimensions of variability yielded the four exemplar sequences used.

GROUNDS → SUGAR (PACK) → CREAM → DRINK
GROUNDS → SUGAR (BOWL) → CREAM → DRINK
GROUNDS → CREAM → SUGAR (PACK) → DRINK
GROUNDS → CREAM → SUGAR (BOWL) → DRINK

The hierarchical structure of everyday tasks means that subtasks can appear as part of different overall tasks. In order to pose this challenge to the model, a secondary task was added to the training set. The task chosen—tea-making—is detailed in Table 3. (A second version of the task, not shown, involved adding sugar from a sugar bowl rather than from a packet.) Some features of the tea task will be relevant to our simulations of action errors. For present purposes, we note only that tea-making was implemented so as to contain versions of sugar-adding identical to those involved in coffee-making.

Note that the tea and coffee tasks begin with precisely the same perceptual input. This raised the issue of how the network was to 'decide' which task to perform when this input was presented. As explained further below, in some simulations the choice of task was left to the network. In other simulations, however, it was useful to have some means of directing the network to perform one task or the other. To this end, two additional input units were included: instruct-coffee and instruct-tea. These were intended to represent simply another form of perceptual input. While they can be thought of as representing auditory verbal commands, we included them with the visual input units, thinking of them as representing the visual cue cards used to instruct patients in some of the experiments of Schwartz et al. (1998). When used during training and testing, the instruction units were activated along with the initial perceptual input, and then inactivated. That is, the instruction units were only active during the first cycle of processing. Thus, while they provided an initial cue for action, they could bear none of the burden of representing task context on subsequent cycles of processing.

In addition to the tea and coffee tasks, the network was trained on an additional group of 267 single-step examples we refer to as the *background training set*. This contained one training example for each physically realizable configuration of the coffee-making environment. For each such input, the corresponding output included every action that might plausibly be performed in that environmental context, with unit activations normalized to sum to 1.0. The purpose of adding the background set to the training corpus was to provide the network with information that might otherwise be derived from exposure to a wide variety of routines other than coffee- and tea-making, including information about basic physical constraints and affordances associated with objects.

Model Architecture. The model architecture followed that shown in Figure 3, with a single input and output unit for each of the features and actions listed in Table 2, and 50 hidden units. Recurrent connections were associated with a conduction delay of one time step, thereby instantiating a simple recurrent network (Elman, 1990; Jordan, 1986b). Unit acti-

[3] Localist action representations were employed as a matter of convenience; the paradigm does not require them. Further simulations, to be described elsewhere, coded actions multidimensionally, based on features derived from empirical studies (e.g., Klatzky, Pellegrino, McCloskey, & Lederman, 1993).

Table 1
The Coffee Task—One of Four Versions Used in Training

Step	Fixated object	Held object	Action
1	cup, 1-handle, clear-liquid	nothing	fixate-coffee-pack
2	packet, foil, untorn	nothing	pick-up
3	packet, foil, untorn	packet, foil, untorn	pull-open
4	packet, foil, torn	packet, foil, torn	fixate-cup
5	cup, 1-handle, clear-liquid	packet, foil, torn	pour
6	cup, 1-handle, brown-liquid	packet-foil-torn	fixate-spoon
7	spoon	packet, foil, torn	put-down
8	spoon	nothing	pick-up
9	spoon	spoon	fixate-cup
10	cup, 1-handle, brown-liquid	spoon	stir
11	cup, 1-handle, brown-liquid	spoon	fixate-sugar
12	cup, 2-handles, lid	spoon	put-down
13	cup, 2-handles, lid	nothing	pull-off
14	cup, 2-handles, sugar	lid	fixate-spoon
15	spoon	lid	put-down
16	spoon	nothing	pick-up
17	spoon	spoon	fixate-sugarbowl
18	cup, 2-handles, sugar	spoon	scoop
19	cup, 2-handles, sugar	spoon-sugar	fixate-cup
20	cup, 1-handle, brown-liquid	spoon-sugar	pour
21	cup, 1-handle, brown-liquid	spoon	stir
22	cup, 1-handle, brown-liquid	spoon	fixate-carton
23	carton, closed	spoon	put-down
24	carton, closed	nothing	pick-up
25	carton, closed	carton, closed	peel-open
26	carton, open	carton, open	fixate-cup
27	cup, 1-handle, brown-liquid	carton-open	pour
28	cup, 1-handle, light-, brown-liquid	carton-open	fixate-spoon
29	spoon	carton-open	put-down
30	spoon	nothing	pick-up
31	spoon	spoon	fixate-cup
32	cup, 1-handle, light-, brown-liquid	spoon	stir
33	cup, 1-handle, light-, brown-liquid	spoon	put-down
34	cup, 1-handle, light-, brown-liquid	nothing	pick-up
35	cup, 1-handle, light-, brown-liquid	cup, 1-handle, light-, brown-liquid	sip
36	cup, 1-handle, light-, brown-liquid	cup, 1-handle, light-, brown-liquid	sip
37	cup, 1-handle, empty	cup, 1-handle, empty	say-done

vations were a smooth, nonlinear (logistic) function of their summed input from other units,

$$a_j = \frac{1}{1+\exp(-\sum_i a_i w_{ij})}$$

where a_j is the activation of unit j, w_{ij} is the weight on the connection from unit i to unit j, and $\exp(\cdot)$ is the exponential function. An environmental feedback loop was implemented as described earlier, using the Perl scripting language (Wall, Christiansen, & Orwant, 2000).

Training Procedure. The training set included all four versions of the coffee task, both versions of the tea task, and the entire set of background examples. Each of the four coffee sequences occurred twice during each epoch (pass through the training set): once with the instruct-coffee unit, and once without. Each tea sequence appeared four times, twice with the instruct-tea unit and twice without.

The network was trained on the target sequences using a version of recurrent back-propagation through time, adapted to the SRN architecture (see Williams & Zipser, 1995). Connection weights were initialized to small random values

Table 3
The Tea Task—One of Two Versions

Step	Fixated object	Held object	Action
1	cup, 1-handle, clear-liquid	nothing	fixate-teabag
2	teabag	nothing	pick-up
3	teabag	teabag	fixate-cup
4	cup, 1-handle, clear-liquid	teabag	dip
5	cup, 1-handle, brown-liquid	teabag	fixate-sugar
6	packet, white-paper, untorn	teabag	put-down
7	packet, white-paper, untorn	nothing	pick-up
8	packet, white-paper, untorn	packet, white-paper, untorn	tear-open
9	packet, white-paper, torn	packet, white-paper, torn	fixate-cup
10	cup, 1-handle, brown-liquid	packet, white-paper, torn	pour
11	cup, 1-handle, brown-liquid	packet, white-paper, torn	fixate-spoon
12	spoon	packet, white-paper, torn	put-down
13	spoon	nothing	pick-up
14	spoon	spoon	fixate-cup
15	cup, 1-handle, brown-liquid	spoon	stir
16	cup, 1-handle, brown-liquid	spoon	put-down
17	cup, 1-handle, brown-liquid	nothing	pick-up
18	cup, 1-handle, brown-liquid	cup, 1-handle, brown-liquid	sip
19	cup, 1-handle, brown-liquid	cup, 1-handle, brown-liquid	sip
20	cup, 1-handle, empty	cup, 1-handle, empty	say-done

(sampled uniformly between ± 1). At the beginning of each training sequence, activations over the hidden units were initialized to random values (sampled uniformly between 0.01 and 0.99). This random initial state was intended to serve as a proxy for internal states that would be present if the model were involved in a variety of other activities just prior to entering the coffee- or tea-making task. On each time step during training, an input pattern, corresponding to a particular combination of viewed and held objects was applied to the input layer of the network (see Figs. 1 and 3). Activation was allowed to propagate through the network, producing a pattern of activation over the output layer. This output pattern was compared with the correct output for that step in the sequence (as defined by the target sequence). The difference between these two patterns, measured by their *cross-entropy* (see Hinton, 1989), was employed as a measure of performance error, providing the basis for gradual, adaptive weight changes. Note that, during training, the correct input sequence was presented to the network regardless of its generated outputs.

Weight updates were performed at the end of each individual sequence, using a learning rate of 0.001 and no momentum. Training was stopped after 20,000 passes through the training set, a point at which error levels had plateaued.[4]

Earlier work has shown that SRNs can have difficulty learning to preserve contextual information through sequences in which it is not immediately useful in selecting outputs (e.g., Servan-Schreiber, Cleeremans, & McClelland, 1988). In order to support this aspect of learning, we introduced an additional term into the network's performance measure that pressured hidden unit activations to change as little as possible over successive processing steps.[5] Although this learning constraint was employed in all of the simulations reported here, we found in subsequent simulations that dropping the constraint yielded equivalent results.

Testing Procedure. The model was tested by allowing it to produce sequences of actions without the external feedback provided during training. Prior to each test run, a random

[4] Later investigations revealed that correct responses, based on a winner-take-all criterion, were produced after about half as many epochs. Even given this, learning in the model may appear surprisingly slow. In this regard, it should be noted that learning required the network to acquire both task knowledge and background knowledge simultaneously. If the network begins the learning process having already encoded relevant background knowledge, task acquisition can be more rapid, especially if the network has been exposed to tasks that share structure with the target task (for relevant simulations, see Botvinick & Plaut, in press).

[5] In back-propagation, weight changes are made based on how they affect output error which, in turn, depends on how they affect unit activations. Thus, the procedure includes a computation of the derivative of error with respect to each unit's activation on each cycle. Our modification involved adding the term $2\rho(a(t) - a(t-1))$ to the derivative for the network's hidden units, where a is the unit's activation and ρ is a scaling parameter (set to 0.05 in our simulations). This imposes a penalty on changing hidden unit activations that scales as the square of the difference between each hidden unit's activation on the present cycle and its activation on the previous one.

pattern of activation was applied to the hidden layer, as during training.[6] On the first cycle of processing, the initial input pattern from the coffee and tea sequences was applied. Once activation had propagated to the output layer, the most active output unit was taken to specify the action selected. The representation of the environment was updated based on this action (even if incorrect), and used to generate a new input pattern.

A total of 300 test runs were completed. In 100 of these, the instruct-coffee unit was included in the initial input pattern, in the next set of 100 the instruct-tea unit was included, and in the third set of 100 no instruction unit was activated. On each test trial, output was collected until the say-done action was produced.

Evaluation of Performance. The goal of the present simulation was to establish the network's capacity to acquire and reproduce the full set of sequences included in the training corpus. Evaluation of performance was straightforward: The sequences produced at test were categorized based on the specific target sequences with which they matched, with a residual error category reserved for sequences not precisely matching any target sequence.

Table 4
Sequences Produced by the Model in the Absence of Noise

With coffee instruction	
GROUNDS → SUGAR (PACK) → CREAM → DRINK	35
GROUNDS → SUGAR (BOWL) → CREAM → DRINK	37
GROUNDS → CREAM → SUGAR (PACK) → DRINK	14
GROUNDS → CREAM → SUGAR (BOWL) → DRINK	14
ERRORS	0
With tea instruction	
TEABAG → SUGAR (PACK) → DRINK	46
TEABAG → SUGAR (BOWL) → DRINK	54
ERRORS	0
With no instruction	
GROUNDS → SUGAR (PACK) → CREAM → DRINK	15
GROUNDS → SUGAR (BOWL) → CREAM → DRINK	18
GROUNDS → CREAM → SUGAR (PACK) → DRINK	12
GROUNDS → CREAM → SUGAR (BOWL) → DRINK	10
TEABAG → SUGAR (PACK) → DRINK	20
TEABAG → SUGAR (BOWL) → DRINK	25
ERRORS	0

Results

When the fully trained model was permitted to select actions without the feedback provided during training, it reproduced, without errors, each of the sequences from the training corpus. On each test run initiated with the instruct-coffee unit, the model produced one of the four coffee-making sequences. The number of occurrences of each variant of the task, in 100 test runs, is shown in Table 4. On each test run using the instruct-tea unit, the model produced one of the two tea sequences, as also shown in the table.

Importantly, production of the target sequences did not require inclusion of the instruction units. On test runs where no instruction unit was activated, the model produced one of the six training sequences, with the frequencies shown in Table 4.

Analyses

Although the coffee and tea tasks may appear simple, it is worth emphasizing the fact that, together, they posed to the network the full set of computational challenges that hierarchical models have traditionally addressed, as enumerated in our list of benchmarks. The network's performance demonstrates its ability to cope with situations where subtasks can be executed in variable order, situations where different versions of one subtask can be substituted for one another, and situations involving hidden environmental states (here, whether sugar has been added to the cup). Above all, it is worth emphasizing the tasks' hierarchical structure. The simulation results demonstrate that such structure can be managed by a processing system that is not explicitly hierarchical in form. Rather than expressing the hierarchical organization of the task domain at the architectural level, the model captures this structure in the internal representations it uses to perform the task. Because these representations are the key to understanding the behavior of the model, it is worth considering them in some detail.

Internal Representations. On each step of processing, the model's hidden layer assumes a new pattern of activation. Since this pattern reflects both the current stimulus-response mapping and any contextual information being preserved, it can be thought of as a compact and context-specific representation of the current step in the task. As there are fifty units in the hidden layer, this internal representation can be represented as a point in a fifty-dimensional state space. As the task sequence unfolds and the model's internal representation evolves, a trajectory is traced through that space. As Elman (1991, 1993) has shown, it is useful to visualize such trajectories as a way of understanding how recurrent models represent tasks they have learned to perform. One way of accomplishing this is through multidimensional scaling (MDS). MDS yields a representation in two dimensions that preserves as much information as possible about the original distances among a set of points in a higher dimensional space (Kruskal & Wish, 1978).

An illustration of MDS applied to the present model's internal representations is presented in Figure 5. This shows the trajectory followed during the entire coffee-making sequence, GROUNDS → SUGAR (PACK) → CREAM → DRINK. The individual points represent the hidden representations arising on each of the 37 steps of the sequence. The line segments connecting the points indicate their arrangement in time. The sequence begins to the far right (at the 'o') and

[6] Comparable performance was obtained in simulations where the hidden layer was not reset between trials.

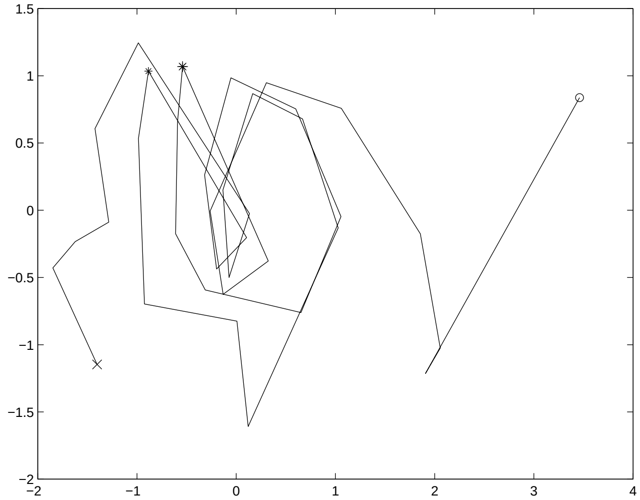

Figure 5. MDS plot of hidden representations over the course of the sequence GROUNDS → SUGAR (PACK) → CREAM → DRINK. The characters ‘o’ and ‘x’ indicate beginning and end of sequence, respectively (and this convention applies throughout the following MDS diagrams).

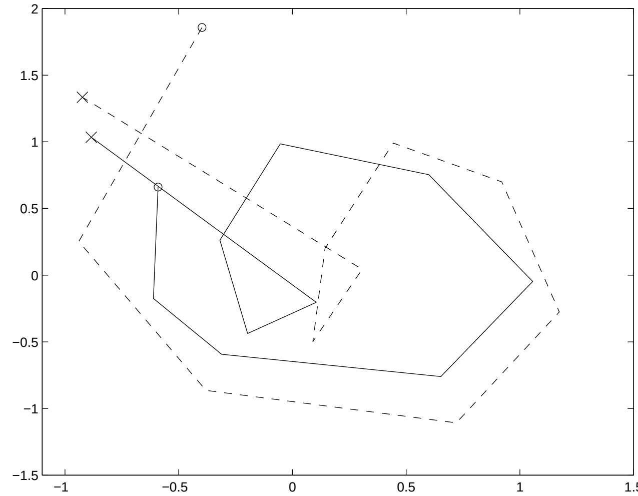

Figure 6. MDS plot of the hidden representations arising during the sugar-packet sequence as performed in the context of coffee-making (solid) and tea-making (dashed).

ends on the far left of the diagram (at the ‘x’).

MDS provides an indication of how the network is able to cope with hierarchical task structure. As an initial example, consider the two points labeled in the diagram with an asterisk. These points represent the patterns used by the model in producing the action stir, in the context of two different subtasks (GROUNDS and SUGAR (PACK)). The points are close together, indicating that the internal representations used on these two steps are similar to one another. This is reasonable, given that the network is producing the same action output in the two cases. However, the plot also indicates that the patterns used on the two steps are not quite identical. The separation between the two points indicates that the network is using slightly different patterns in order to produce the stir action. The model’s internal representation are “shaded” (a term suggested by Servan-Schreiber et al., 1991) in a way that carries information about the larger temporal context in which individual actions are being performed.

The same mechanism allows the network to deal with the fact that entire subtasks can appear in different task contexts. An example here is the sugar-adding subtask. Because sugar-adding appears as part of both coffee- and tea-making, the network must maintain a representation of the overall task context as it performs the subtask. An indication of how this is accomplished is provided by Figure 6. This shows an MDS plot of the internal representations arising during performance of sugar-adding in the settings of coffee-making (solid trajectory) and tea-making (dashed). The first point to note is that the two trajectories are closely aligned. This reflects the fact that, across the two contexts, the network uses similar representations for each step of the subtask. This is not surprising, given that the subtask is performed in precisely the same way in the two tasks. However, while the trajectories are similar, they are not perfectly identical. The slight differences between the two indicates that, at each step of the subtask, identical perception-action mappings are being represented slightly differently, depending on the overall task context. Here, as in the previous example, the network’s internal representation on each step is shaded to reflect differences in the larger task setting.

Representational Shading vs. Schema Nodes. Because this representational shading is the means by which the present model represents temporal or task context, it can be seen as a functional analogue of the schema nodes employed in hierarchical models. However, there are a number of differences between the two. Three are particularly worth noting.

1. The memory mechanism at work in the present model is capable of preserving specific information about earlier actions, and about hidden states of the environment. For example, during coffee-making, the model’s hidden representations are shaded to reflect whether sugar has, or has not, yet been added (Figure 7). In hierarchical models, preserving such information requires separate mechanisms (i.e., reflex inhibition and symbolic gating).

2. Because the present model employs distributed representations, it can respond to the similarity relations among actions or routines. As shown in Simulation 1a below, this allows the network to implement a form of information-sharing in representing inter-related tasks. As explored in Simulations 2 and 3, this feature of the model also contributes to its tendency to make certain kinds of error.

3. Hierarchical models, by definition, represent different levels of task structure in a disjunctive, non-overlapping fashion. There is no such restriction in the present model; information pertaining to task, subtask, action (and intermediate)

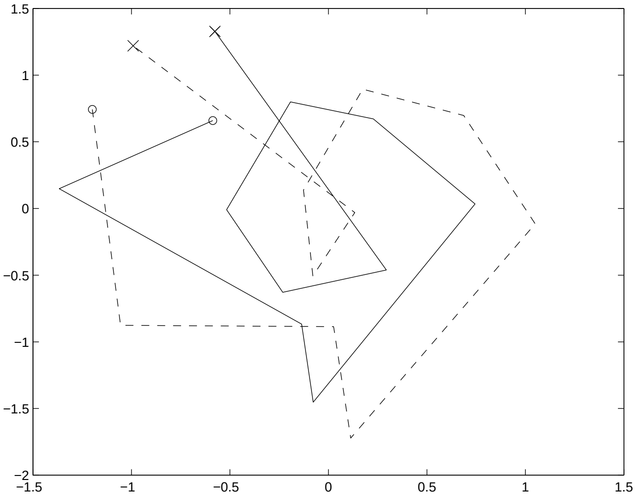

Figure 7. MDS representation of the hidden representations arising during performance of the cream-adding sequence, as performed before (solid) and after (dashed) sugar-adding.

levels can combine and overlap in any way necessary to support task performance. The implications of this point become clear in task domains that are not strictly hierarchical. We examined the model's performance in one such domain in the following simulation.

Simulation 1a: Performance on a Quasi-Hierarchical Task

One characteristic of normal routine action that we have emphasized is its context dependence. This refers to the way in which the details of a task or subtask vary depending on the particular situation in which it is performed. In our basic implementation of the coffee task, such context dependence is largely filtered out. The steps followed in adding cream, adding sugar, drinking, etc. are the same regardless of the temporal context in which they occur. In order to address the issue of context dependence, we therefore carried out a separate simulation. Here, the model used in Simulation 1 was retrained using a set of tasks intended to implement the example given in the Introduction, involving a waiter preparing coffee for three regular customers, each of whom expects a different amount of sugar.

Methods. Three new instruction units were added to the basic model, corresponding to requests for coffee with no sugar, with one spoonful of sugar, and with two spoonfuls, respectively. The model was trained in the same manner as in Simulation 1, but using a training set containing modified versions of the coffee sequence shown in Table 1. Each sequence began with the initial input used in that sequence, but now accompanied by one of the new instruction units. The sequence for making coffee without sugar followed the original coffee sequence, but omitted the entire sugar sequence. The sequence for making coffee with one spoonful of sugar was drawn directly from the original set of four coffee sequences (GROUNDS → SUGAR(BOWL) → CREAM → DRINK). The sequence for making coffee with two spoonfuls of sugar was identical to the previous sequence, but the sequence fixate-sugarbowl → scoop → fixate-cup → pour appeared twice in succession, rather than being performed once as in the original version of the task. In view of the fact that no variability in performance was required of the model, the hidden layer was initialized by setting all unit activations to 0.5, rather than to random values. The training and testing procedures were otherwise identical to those used in Simulation 1. As in that simulation, training was terminated when it was evident that learning had reached a final plateau (20,000 epochs).

Results and Analysis. Testing of the fully trained model indicated that it was able to perform all three versions of coffee-making without errors. In accordance with the instruction unit activated on the first cycle of processing, the network reproduced the appropriate version of the coffee-making sequence from the training set, either omitting sugar or adding one or two spoonfuls from the sugar bowl.

The model's method of dealing with the sequences involved in this simulation provides a contrast to traditional, hierarchically structured models of sequential action. As discussed in the Introduction, such models face an uncomfortable choice between representing different variations of a sequence using a single unit or using multiple independent units. The present model, in contrast, provides a natural means for simultaneously encoding the relatedness of two sequences while at the same time representing their differences. Once again, the point is made clear by an examination of the internal representations the model uses during task processing. Figure 8 shows a set of MDS plots based on the patterns of activation arising in the model's hidden layer, on each step of processing, in each of the sequences in the training set. What the figure indicates is that there is a family resemblance among the internal representations the model uses across the three versions of the task. Within each row of the figure, the representations trace out roughly similar trajectories, reflecting the fact that the model has picked up on the similarity among the sequences the tasks involve. In the case of sugar-adding, where the network is performing different versions of the subtask, the resemblance between the two trajectories indicates that the network has capitalized on the presence of shared structure; where there are steps shared by the two versions of sugar-adding, very similar internal representations are used for each.

In this simulation, as in Simulation 1, a key to the model's performance is its ability to capture, in a single representation, both the similarities and differences between task contexts. It is this property that permits the model to represent multiple levels of structure simultaneously, to preserve and exploit information about previous actions, and to deal with quasi-hierarchical task structure. However, while this feature of the model gives it its computational power, it also makes the model vulnerable to errors involving confusions between task contexts. Simulations 2 and 3 examine this aspect of the

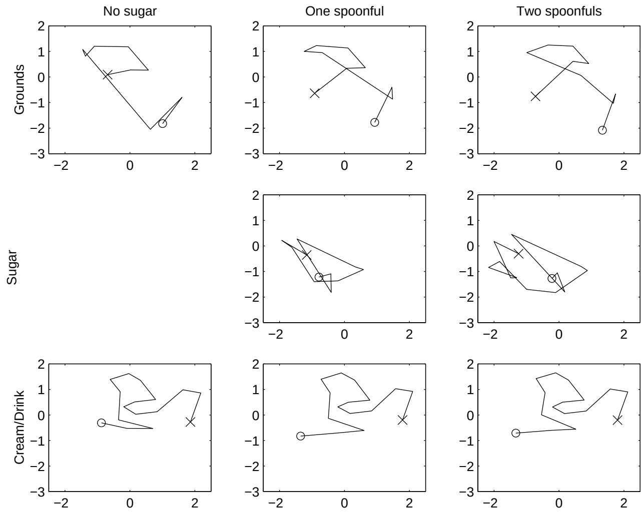

Figure 8. MDS plots of the internal representations underlying the model's performance in the waiter scenario. The sequence in which no sugar was added is shown in the left column, the sequence with one spoonful of sugar in the middle column, and the sequence with two spoonfuls of sugar in the right column. The steps involved in the sugar-adding sequence itself are shown in the middle row of panels. The sequence of steps preceding sugar-adding are shown in the top row, and those following it in the bottom row.

model.

Simulation 2: Slips of Action

In this simulation, we examined the ability of the model studied in Simulation 1 to account for several basic characteristics of human slips of action. Slips are the "absent-minded" mistakes normal individuals make from time to time while performing familiar tasks. While such errors have been of interest to psychologists since William James (1890, see also Jastrow, 1905), the most detailed information about the timing and form of such errors comes from work by James Reason (1979, 1984a, 1984b, 1990, 1992). Reason conducted diary studies in which subjects recorded the details of their own slips of action. In addition to Reason's own interpretation of this data, it has also been analyzed and extended by Norman (1981) and others (e.g., Baars, 1992; Roy, 1982). Such work points to a number of general principles that any sufficient model should address.

1. As noted earlier, slips tend to occur under conditions of distraction or preoccupation (Reason, 1990).

2. Slips tend to occur at "branch points" or "decision points," junctures where the immediately preceding actions and/or the environmental context bear associations with different subsequent actions. The following example is offered by Reason (1990, p. 70): "On passing through the back porch on my way to get my car I stopped to put on my Wellington boots and gardening jacket as if to work on the garden." Here, the behavioral and physical contexts relate to more than one activity, creating what Reason calls a branch point. The error involves a figurative (as well as literal) wrong turn at this point.

3. In the phenomenon Norman (1981) labeled "capture," lapses from one task into another tend to occur just after a series of actions that the two tasks share. The literature on action slips provides a wealth of examples of such errors, such as Norman's (1981) lapse from page-counting to card-counting or James' (1890) lapse from dressing for dinner into dressing for bed.

4. Rather than involving bizarre or disorganized action sequences, slips tend to take the form of a familiar and intact sequence, ordinarily performed in a different but related context (Reason, 1979, 1984b).

5. Sequencing errors tend to fall into three basic categories: Perseverations, omissions, and intrusions (Reason, 1984a; object substitutions form a fourth major category of slip). Perseverations (or repetitions) occur when the sequence constituting the error derives from earlier within the same task. Intrusions occur when the sequence comes from a different, usually related, task. Omissions involve skipping over a subroutine to execute a sequence from later in the same task (e.g., "I wrote out a cheque and put the cheque book back into my bag without tearing the check out," Reason, 1984b, p. 535).

6. Slips involving lapses from one task into another (i.e., intrusions) tend to reflect the relative frequency of the two tasks. Specifically, lapses tend to involve a shift from a less frequently performed task into one more frequently performed (Reason, 1979).

As introduced earlier, our approach in simulating slips of action was based on the widely shared assumption that slips result from degradation to representations of task context. In the setting of the present simulation, this meant examining the model's performance under conditions that mildly distorted the patterns of activation arising in its internal layer.

Methods

The simulation was conducted using the model described in Simulation 1, complete with its final set of connection weights. However here, in order to simulate the conditions involved in slips of action and ADS, the model's sequencing mechanism was disrupted by adding zero-mean normally distributed random noise to activation values in the hidden layer at the end of each cycle of processing, after the completion of action selection. Test runs were otherwise conducted as in Simulation 1. Each trial began with one instruction unit activated (as before, for the first step of the trial only), and was terminated when the say-done action was selected, or after 100 cycles. Two hundred trials were conducted, half using the instruct-coffee unit and half using instruct-tea, at each of the following levels of noise (variance): 0.02, 0.04, 0.08, 0.1, 0.2, 0.3, 0.4, and 0.5.

Because slips of action are relatively infrequent in human behavior, analysis focused on model behavior at levels of noise producing an error rate of less than 0.5 errors per trial. The approach to evaluating network performance is de-

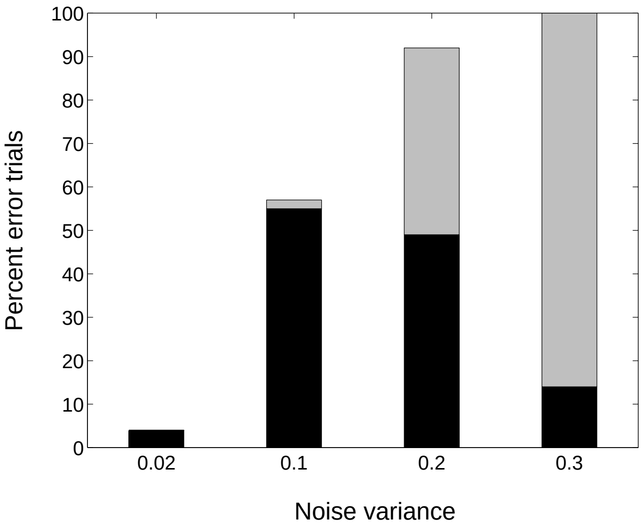

Figure 9. Number of trials in 100 containing at least one error, at four levels of noise (overall bar height). The black portion of each bar indicates trials that contained only errors involving intact but displaced subtasks. The gray area indicates trials involving within-subtask disorganization.

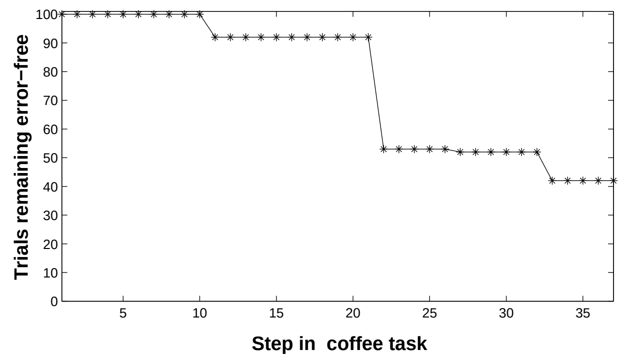

Figure 10. Survival plot for the coffee task under noise with variance 0.1. On the x-axis are steps in the sequence. The y-axis indicates the number of trials, from a total of 100, that remained error-free at the corresponding step. Data is collapsed across the four versions of the task.

scribed in conjunction with the simulation results.

Results

The proportion of errors occurring at each level of noise is shown in Figure 9.[7] Our focus in this section of the study was on error rates lower than 0.5, and thus on noise levels lower than 0.1 (see Simulation 3 for analysis of model performance at higher levels of noise). In this range, in keeping with human behavior, the model's errors tended to occur at so-called branch points, corresponding here to the transitions between subtasks. One way to illustrate this behavior is with a survival plot, as shown in Figure 10. The data diagrammed here are based on 100 test runs on the coffee-making task, applying noise with variance 0.1. The horizontal axis indexes the steps in the task. The vertical axis shows the number of trials for which no error had yet occurred at the corresponding step. Occurrence of errors at a given point is indicated by a sudden step down in the diagram, the size of which reflects the frequency of errors at that step. The plot contains large drops at three specific steps, each of which corresponds to a point in the task where a subtask has just ended and a new one begins.

Also in keeping with empirical findings, the model's errors tended to take the form of recognizable subtask sequences, inserted at the wrong moment, but nonetheless drawn intact from somewhere in the training corpus. In some instances, the inserted sequence came from earlier within the task being performed, resulting in a perseveration or repetition error (e.g., adding sugar twice, either in succession or with intervening subtasks). In other instances, the inserted sequence came from later in the task, resulting in a subtask omission (e.g., leaving out the sugar subtask and skipping directly to cream-adding and then drinking). In still other cases the inserted subtask came from outside the task being performed, resulting in an intrusion error. The prime example here occurred in the context of tea-making. As shown in Table 3, our implementation of the tea task did not include the cream-adding subtask. Thus, if a trial included the tea-bag subtask and, later, cream-adding, this reflected a lapse from tea- into coffee-making. This was, in fact, the network's most common error during tea-making.

The tendency for errors to take the form of intact but displaced subtask sequences is illustrated in Figure 9. The black portion of each bar in this diagram indicates the number of trials at each level of noise that contained *only* errors involving subtask displacement. The gray portion stacked above this shows the number of trials containing at least one error involving a disordered subtask sequence (e.g., fixate-carton → pick-up → peel-open → fixate-cup → put-down). As the figure makes clear, at low overall error rates, errors primarily took the form of intact subtask sequences.

In summary, the errors produced by the model under low levels of noise displayed three principal characteristics of slips of action produced by normal subjects: The errors tended to occur at branch points; they tended to take the form of displaced but well-formed action sequences; and they involved omissions, repetitions, or intrusions.

Analyses

In this simulation, errors resulted from a degradation of the model's representations of task context. The functional consequences of such distortion follow from a basic property of connectionist models: If faced with a novel or distorted representation, such models tend to respond based on that representation's *similarity* to more familiar ones (Rumelhart,

[7] The data shown are from the coffee task. However, similar data were obtained for the tea task, both here and in subsequent analyses.

Durbin, Golden, & Chauvin, 1996). When the present model is faced with a distorted context representation, it responds based on the representation's similarities to the set of canonical representations it has learned to use in performing the target tasks. An error occurs when the network is in a situation *s*, calling for action *a*, but distortion causes its context representation to resemble a pattern the model has learned to associate with a different situation, and a different action.

A Case Study. For illustration, consider the following relatively common error: After adding grounds, sugar, and cream, the model selects fixate-sugar and enters into a second round of sugar-adding, committing a perseveration error. In explicating this error, it is useful to consider how the error is ordinarily avoided. When operating without noise, the model is able to keep track of whether sugar has been added by appropriately shading its internal representations, as shown in Figure 7. When the model reaches the critical juncture, at the end of the cream-adding subtask, it will be in one of the two states indicated with an 'x' in that figure. One of these patterns indicates that sugar has not yet been added, the other that it has. For brevity, we'll label these patterns Cr_{11}^{nosug} and Cr_{11}^{sug}, respectively. In this notation, the main element designates a specific subsequence (e.g., "*Cr*" for adding cream), the subscript designates a specific step in this sequence (e.g., 11) and the superscript designates the larger context in which the sequence occurs (e.g., "nosug" for a point prior to adding sugar). The sugar perseveration error occurs when the model is in the situation usually represented by Cr_{11}^{sug} (i.e., having added cream after sugar), but noise causes the model's context representation instead to resemble Cr_{11}^{nosug} (i.e., having added cream but before adding sugar).

Evidence for this account of the model's behavior is provided in Figure 11. This focuses on the context representations arising at the end of the cream-adding sequence, comparing these between correct trials and trials where a sugar perseveration occurred. For each trial-type, the plot shows the average distance of the actual context representation from the two canonical patterns Cr_{11}^{nosug} and Cr_{11}^{sug}. What the figure indicates is that the internal representations on correct trials, though distorted, still tended to resemble the context-appropriate pattern Cr_{11}^{sug} more closely than Cr_{11}^{nosug}. In contrast, on trials where a sugar perseveration occurred, the context representations present at the point of the error tended to fall closer to Cr_{11}^{nosug}. On these trials, noise caused the model, in effect, to 'forget' that it had previously added sugar.

Branch Points. Critically, when such confusions between contexts do occur, their effect on overt performance is most often felt at branch points. This is because branch points involve a situation where similar context representations are associated with different actions. For example, in the case of the sugar perseveration error we have just been considering, it is important that Cr_{11}^{nosug} and Cr_{11}^{sug} lie close to one another in representational space (see Figure 7). In the setting of mild representational distortion, this similarity makes it easy for the network to 'mistake' one context for the other.

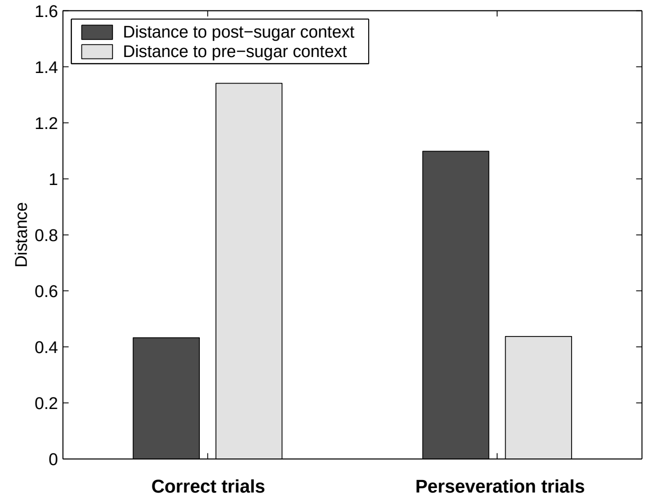

Figure 11. Comparison of the internal representations arising on the last step of the cream subtask, for trials where this step led into a sugar perseveration error and for correct trials. Data for both correct and error trials are based on averages across a sample of size 10.

Interestingly, while context confusions tend to impact overt behavior at branch points, they may nonetheless be present for multiple time-steps prior to a branch point. Indeed, further analysis of the model's internal representations shows that, in most cases, the model's branch-point errors involve a gradual drift in the representation of context, beginning several steps before the branch point.

For illustration, we focus once more on the example of the sugar perseveration. Figure 12 is based on the context representations arising on the steps leading up to this slip (i.e. the steps in the cream-adding sequence). Like the single step addressed in the previous diagram, here the degraded representation on each step (which we'll designate $Cr_i^{\text{sug}*}$) is visualized in terms of its distance from two canonical vectors, each produced by the model on the corresponding step in the absence of noise. The dashed line shows distances from the canonical pre-sugar patterns (Cr_i^{nosug}); the solid line shows distances from a corresponding set of post-sugar reference points (Cr_i^{sug}). Note that the dashed line in the figure curves upward. This indicates that, over the steps leading up to a sugar perseveration error, the patterns $Cr_i^{\text{sug}*}$ tend to drift gradually away from the canonical patterns Cr_i^{sug}; that is, the network gradually 'forgets' that sugar has already been added. The solid line falls over the course of the cream sequence, showing $Cr_i^{\text{sug}*}$ drifting toward Cr_i^{nosug}. Eventually, the solid and dashed lines cross, representing the point at which the network has effectively forgotten that sugar has been added. Note that, on average, this cross-over occurs several steps prior to the step on which the error actually occurs. The confusion remains latent until the branch point simply because it is only at that point that the particular piece of information that has been corrupted becomes relevant to action selection.

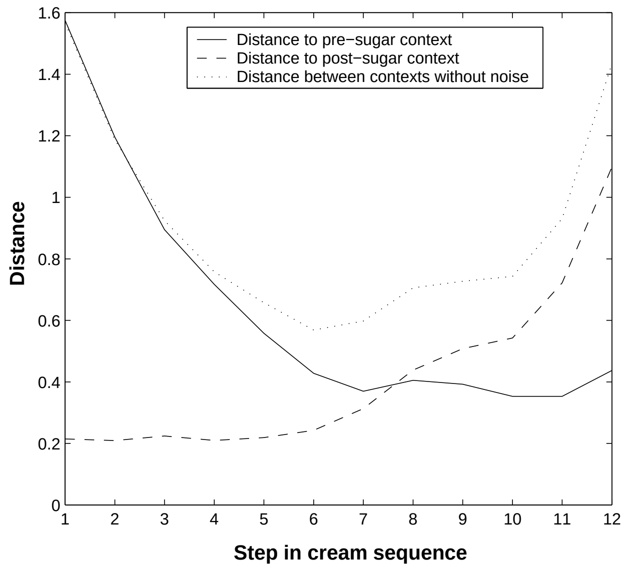

Figure 12. The steps of the cream-adding sequence, leading up to a sugar perseveration error. Solid trace: distance of the average context pattern from the pre-sugar reference for the corresponding step. Dashed: distance from the post-sugar reference. Dotted: distance between the two reference vectors. Data based on averages across a sample of size 10.

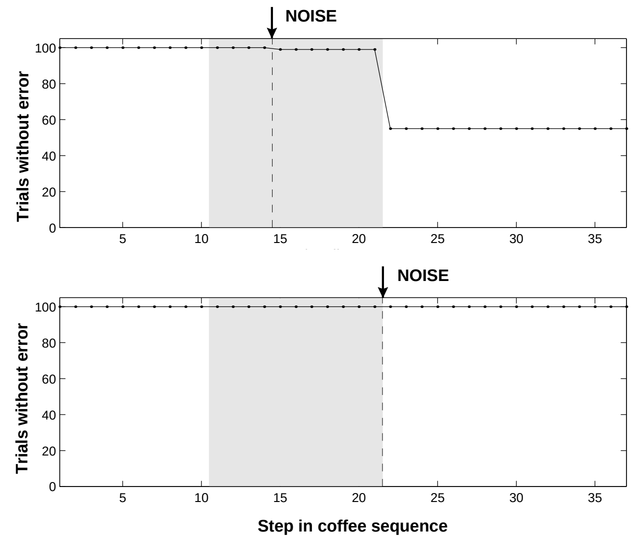

Figure 13. Survival plots, based on 100 coffee-making trials with noise (variance = 0.5) applied at two points in processing. Dashed lines indicate point of noise injection. Steps belonging to the cream-adding subtask are set off in gray. Top: injection of noise at mid-subtask (before step 15). Bottom: injection of noise at subtask end (before step 22).

One reason that branch-point errors are usually associated with a gradual representational drift beginning several steps earlier is that, at low levels of noise, only small distortions occur on each step. It thus requires several incremental distortions to sufficiently disrupt the system's function. However, a further examination of the model's context representations indicates that there is also another reason: It is easier for the model to confuse one context with another when processing is toward the middle of a subtask sequence than when it is near the beginning or end of one. To explain why this is so, we return to the data diagrammed in Figure 12. Recall that the distance data in this figure were computed using the two sets of canonical patterns, Cr_i^{nosug} and Cr_i^{sug}. While the context representations occurring under noise drift toward and away from these reference patterns, it is interesting to note that the reference patterns themselves vary in their distance from one another over the course of the cream-adding sequence. Specifically, as illustrated by the dotted line in Figure 12, the patterns in Cr_i^{nosug} become progressively more similar to their siblings in Cr_i^{sug} during the first half of the sequence, and this trend reverses during the second half of the subtask. The pattern illustrated in the figure means, in effect, that the network represents the pre- and post-sugar situations more similarly toward the center of the subtask sequence than near the branch points at either end. As a result, the center of the subtask represents a point in processing where contextual information is particularly vulnerable to the effects of noise. It is particularly easy here for noise to alter a post-sugar context pattern so that it resembles the corresponding pre-sugar pattern, or vice versa. At the beginning and end of the subtask, where the standard context patterns are more distinct, noise is less likely to have this effect.

Despite its subtlety, this aspect of the model is important, for it leads to a novel prediction about human slips of action. The prediction concerns the impact of momentary distraction, based on the timing of such distraction with respect to subtask boundaries. Specifically, the prediction is that distraction toward the middle of a subtask sequence should give rise to more frequent errors at the transition to the next subtask than distraction closer to that transition point.

The prediction is illustrated in Figure 13. This shows data drawn from simulations in which momentary distraction was modeled by degrading the model's context representation on a single step of processing. Introducing noise at a step near a subtask boundary yielded no errors. However, the same amount of noise injected toward the middle of a subtask sequence resulted in a large number of subsequent errors (occurring at the nearest subsequent subtask boundary). This differential effect of distraction, based on its timing, constitutes a strong prediction of the model, one that appears to differentiate it from previous models of action.

Simulation 2a: Effect of Relative Task Frequency

One particularly interesting aspect of the empirical data concerning intrusions is that these errors show an effect of relative task frequency; intrusions tend to involve a lapse from a less frequently executed task into a more frequent one. Simulation 2 showed that, when the model's context representations are degraded, intrusions are among the errors that it produces. In particular, the model was prone to the error of adding cream to tea, indicating an intrusion from coffee-making into tea-making. The present simulation asked whether the frequency of this intrusion error would vary, like human lapse errors, with relative task frequency. This was tested by retraining the model on training sets involving three different proportions of coffee- and tea-making.

Methods. The model from Simulation 1 was retrained, using the same procedure as in that simulation, but using three modified training sets. The first set included five times as many instances of coffee-making as tea-making, the second equal proportions of the two tasks, and the third five times as many instances of tea-making. The total number of target sequences (coffee plus tea) was balanced across sets.[8] Each training set included the same group of background examples that appeared in the original training set. Training was terminated after it was evident that a final plateau in error had been reached (5000 epochs).

Following training, each version of the model was tested in the standard fashion, using the instruct-tea unit on the first cycle of processing and noise with a variance of 0.1. Evaluation of the model's performance following training on each corpus focused on the frequency of lapses from tea- into coffee-making, indicated by the error of adding cream to tea. Specifically, we asked whether the frequency of this error would vary inversely with the relative frequency of tea-making during training.

Results and Analysis. In accordance with empirical data, the model's behavior did show an effect of task frequency on the tendency to lapse from one task into another. Figure 14 shows the frequency of the cream-into-tea error following training on the three example sets. The lapse error occurred more frequently as the tea task became less frequent in training.

The mechanism behind these results can be understood in much the same terms as those of Simulation 2. Like the errors discussed earlier, the cream-into-tea error occurs when distortion to the context representation leads it to resemble another pattern that is part of the network's repertoire, but which is associated with a different action. As shown in Figure 6, on the last step of the sugar subtask, the model uses one pattern when performing the tea task (call it Sug_{11}^{tea}) and another, slightly different pattern when performing the coffee task (Sug_{11}^{cof}). The cream-into-tea error occurs when the model's internal representation should be Sug_{11}^{tea}, but noise causes it to more closely resemble Sug_{11}^{cof}. Returning to the spatial metaphor, one can imagine the effect of noise as a movement of the model's context representation away from the tea-making reference point, into the vicinity of the coffee-making reference point.

Figure 15 (top) illustrates how this movement affects action selection. The data shown here were produced by instating, and holding constant, the environmental input nor-

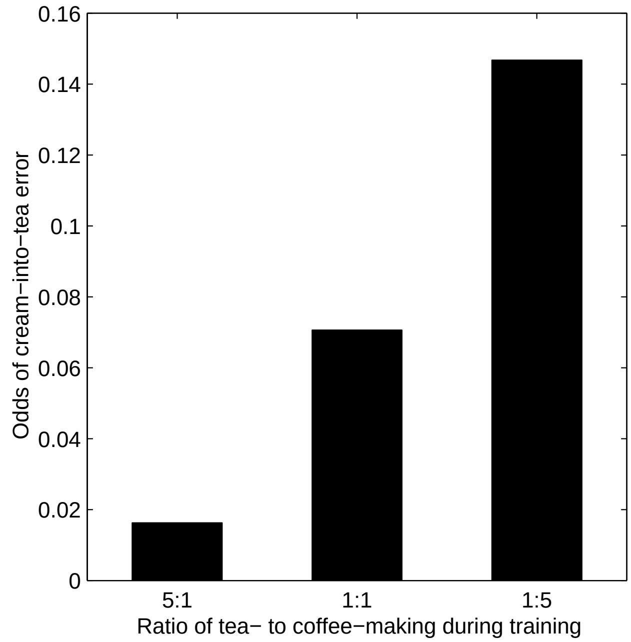

Figure 14. Odds of the cream-into-tea lapse, following training involving varying proportions of coffee- and tea-making. Each data point is based on 500 trials, using a noise level of 0.1.

mally present at the end of the sugar-packet sequence in the tea task, while applying a series of gradually varying context representations. These were produced by starting with Sug_{11}^{tea} and gradually distorting it in the direction of Sug_{11}^{cof}. The effects of this gradual transformation are plotted from left to right in the figure. The data themselves relate to the activation of two output units, put-down, the correct action in the tea context, and fixate-carton, the correct action in the coffee context. As the context pattern diverges from Sug_{11}^{tea}, the network activates the put-down action less strongly, and as the pattern comes to resemble Sug_{11}^{cof}, the network more strongly activates fixate-carton.

The point at which the two traces in the figure intersect provides an indication of the distance the context pattern must travel before the cream-into-tea error will occur. As illustrated in the center and bottom panels of Figure 15, variations in relative task frequency influence where this cross-over occurs along the path from the tea pattern to the coffee pattern. When tea-making occurs more frequently during training, the cross-over point lies further from Sug_{11}^{tea}, meaning that a more severe distortion is required to produce the cream-into-tea error. As a result, the error occurs less frequently. Conversely, when tea-making is relatively infrequent during training, the cross-over point lies closer to Sug_{11}^{tea}. This means that a smaller distortion will cause the error, explaining why it occurs more frequently.

[8] Similar results were obtained in simulations in which the absolute number of presentations of the tea task during training was held constant.

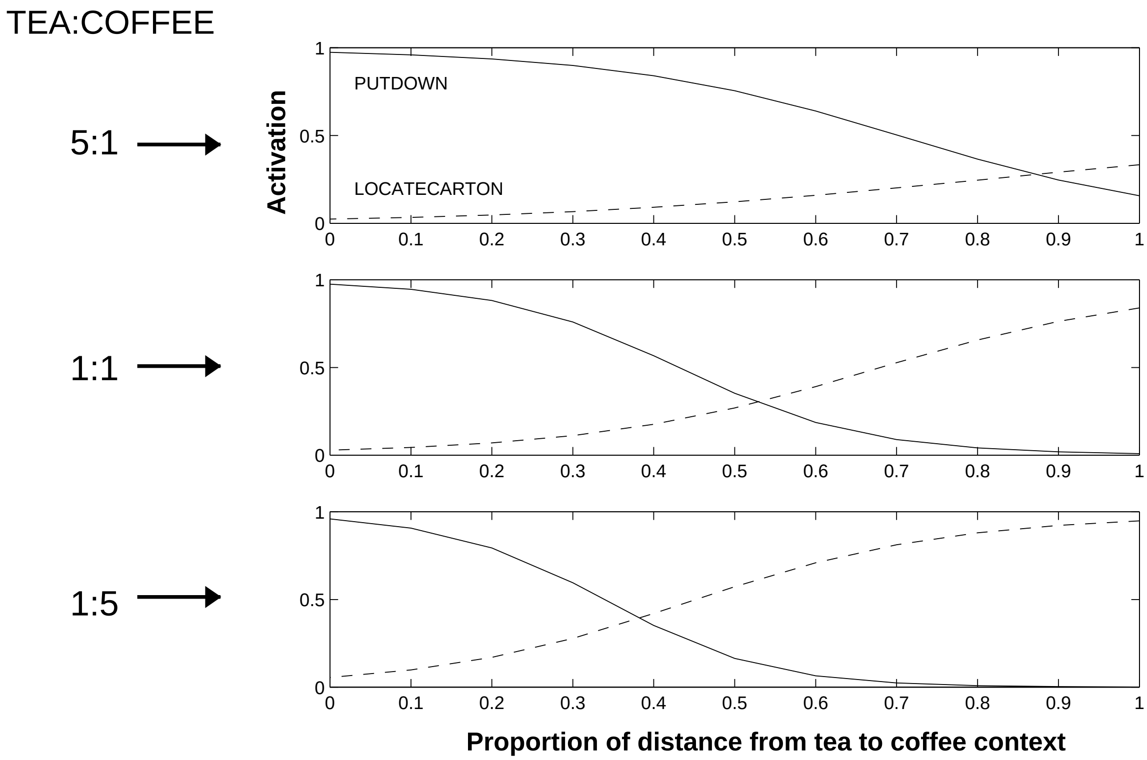

Figure 15. Effect on action selection of progressively distorting the context representation at the end of the sugar sequence in the tea task toward the corresponding coffee context. Ratios of tea to coffee during training: top, 5:1; middle, 1:1; bottom, 1:5.

Simulation 3: Action Disorganization Syndrome

In Simulation 2, mild degradation of the model's representation of temporal context led to errors resembling everyday slips of action. The next segment of the study tested the prediction that more severe degradation would lead to behavior resembling that of patients with ADS.

Impairment in performing everyday sequential routines, especially those involving the use of multiple objects, is frequently observed following brain damage. Most relevant to the issues under investigation here are two closely interrelated neuropsychological syndromes: ideational apraxia and frontal apraxia. While ideational apraxia is traditionally associated with left hemisphere lesions and frontal apraxia with prefrontal damage, in both syndromes patients display disorganization in their approach to sequential tasks (Duncan, 1986; Lehmkuhl & Poeck, 1981). In recent years, studies have challenged the specificity of ideational apraxia to left hemisphere damage (Buxbaum, Schwartz, & Montgomery, 1998), and have blurred the distinction between ideational and frontal apraxia. As a result, the anatomically neutral term action disorganization syndrome (ADS) has been adopted by some researchers (Schwartz, 1995).

Recent studies of ADS, most notably those performed by Schwartz and colleagues (Buxbaum et al., 1998; Schwartz, Mayer, Fitzpatrick, & Montgomery, 1993; Schwartz et al., 1998, 1995, see also Humphreys & Forde, 1999), have utilized explicit coding techniques to analyze patients' performance on naturalistic tasks both in the laboratory and in daily life, and have begun to yield a finer-grained picture of such patients' behavior. The principal findings can be summarized as follows:

1. Patients with ADS produce sequential behavior that is more fragmented than that of normals. Specifically, they show a tendency to abandon a subtask before the goal of that subtask has been accomplished. Schwartz et al. (1991) quantify this tendency by counting actions that occur outside the boundaries of completed subtask sequences, calling these "independent" actions. Schwartz et al. (1991) evaluated the frequency of independent actions in the behavior of an ADS patient as he prepared instant coffee in the context of eating breakfast. In the earliest testing sessions, approximately one month after the onset of the disorder, roughly one half of the patient's actions were independents. Continued observations over the ensuing month revealed a gradual reduction in fragmentation as measured by the proportion of independent actions (Figure 16).

2. In addition to showing a general fragmentation of behavior, ADS patients commit frank errors, which fall into a characteristic set of categories. Most common are omission errors. Across studies, omissions have been consistently found to make up approximately 30–40% of ADS patients'

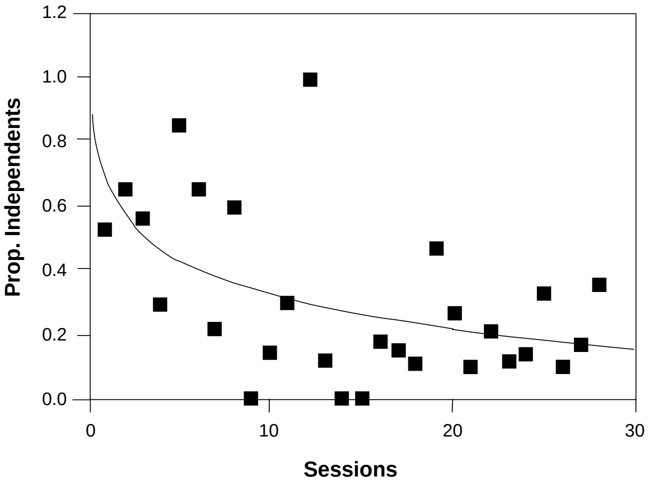

Figure 16. Proportion of independent actions produced by patient HH over the course of his recovery. (From Schwartz et al., 1991)

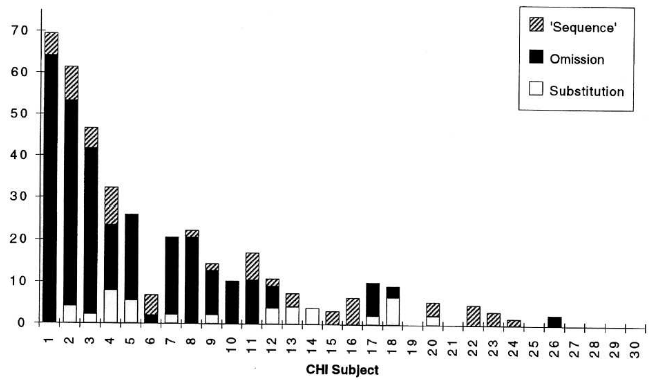

Figure 17. Data from Schwartz et al. (1998), showing the relationship between overall error rate and the distribution of error-types in a group of closed head injury patients. Each bar represents an individual participant.

errors (Buxbaum et al., 1998; Humphreys & Forde, 1999; Schwartz & Buxbaum, 1997; Schwartz et al., 1998). Next most frequent are sequencing errors, including repetitions of either of single steps or entire subtasks; anticipation errors, where an action is undertaken before a prerequisite action is completed, for example pouring a container of cream without having yet opened it; and (rarely) reversal errors, where two steps in a sequence are performed in the incorrect order. Sequencing errors, considered as a group, tend to make up approximately 20% of all errors. Other error types include action additions (actions that do not appear to belong to the assigned task), and substitution errors, where the correct action is performed using the wrong object, or the wrong action is performed with the correct implement. Schwartz et al. (1998) found that substitutions comprised 10% of errors, additions 12%. Less frequent error types observed in ADS include tool omissions (e.g., pouring sugar straight from a sugar bowl; 3% of errors in Schwartz et al., 1998), and quality errors (e.g., pouring far too much sugar into a cup of coffee; 8% of errors).

3. An important observation concerning omission errors in ADS is reported by Schwartz et al. (1998) and Buxbaum et al. (1998). As mentioned earlier, they found that, across patients, the proportion of omissions correlated with overall error rate (Figure 17). In mildly impaired subjects, omissions occurred with about the same frequency as sequence and substitution errors. In contrast, for subjects with the highest error rates, omissions formed a large majority of the errors committed.

Methods

This portion of the study involved a further analysis of the data produced in Simulation 2. In that simulation, the model was tested at multiple levels of noise variance, but analyses focused only on the range producing relatively low error rates (variance 0.1 and below). The present section of the study focused instead on model performance at higher levels of noise (0.1-0.5).

Performance at these noise levels was compared with the behavior of patients with ADS. In order to compare model performance with the data on independents reported by Schwartz et al. (1991), the coding scheme used by these researchers was adapted to our implementation of the coffee-making task, as detailed in the Appendix. Comparison of the specific error-types produced by the model with those of ADS patients was based on the classification specified by Schwartz et al. (1998), which includes object substitutions, gesture substitutions, action additions, tool omissions, quality errors, omission errors, and sequence errors comprising anticipation-omission errors, reversals, and perseverations.[9]

Initial evaluation of the model's errors was directed at establishing whether the model produced examples of each of the above varieties of error. Quantitative analysis focused specifically on sequence and omission errors. Enumeration of these errors was approached using a simplified version of the coding scheme employed by Schwartz et al. (1998), as described in the Appendix.

Results

In keeping with the performance of ADS patients, the sequences produced by the model became increasingly fragmented with increasing noise. At noise levels above 0.1, errors first began to appear within subtask boundaries, rather than only at the transitions between subtasks (see Figure 9). A typical example is shown in Table 5 (left). With increasing noise, sequencing both within and between subtasks became increasingly disrupted (Table 5, center). At high levels of noise, only short fragments of the original subtask sequences could be discerned, as shown in Table 5 (right). At extreme

[9] Spatial misorientation and spatial misestimation errors are inapplicable given our non-spatial coding of action; these error types are also infrequent in the behavior of ADS patients (Schwartz et al., 1998).

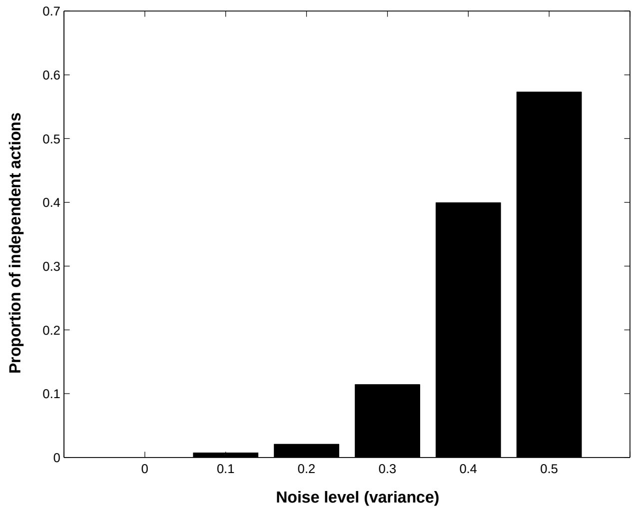

Figure 18. Proportion of independent actions produced by the model at several levels of noise severity.

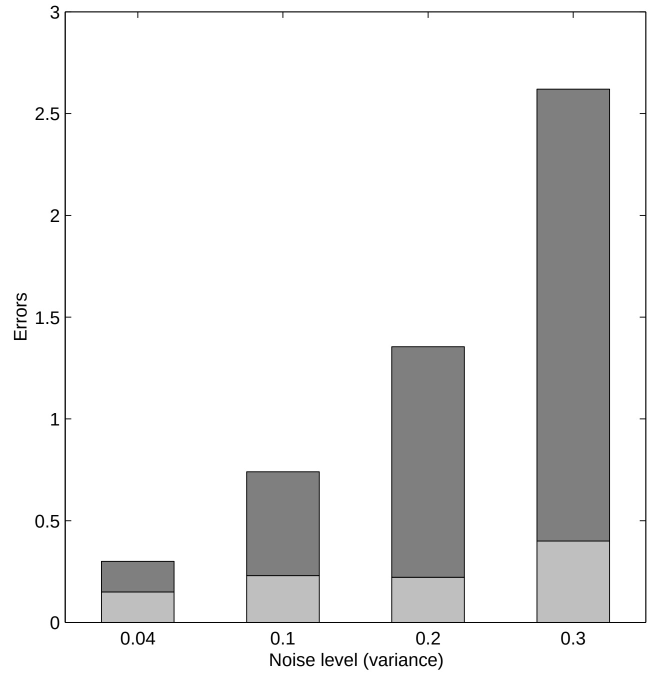

Figure 19. Average number of sequence (light gray) and omission (dark gray) errors per trial at several noise levels. Based on a sample of 100 trials at each noise level.

levels of noise, trials were increasingly taken up with extended periods of rather aimless "toying" behavior, which is also characteristic of the behavior of highly impaired ADS patients (Schwartz et al., 1991).

A rough quantification of the degree to which sequential structure is disrupted is provided by the frequency of independent actions (as defined by Schwartz et al., 1991 and the Appendix). As shown in Figure 18, the proportion of independent actions increased smoothly with the severity of noise. It is significant that the change in fragmentation is graded, since the empirical data show that graded changes in the fragmentation of patient performance can occur over the course of recovery (Schwartz et al., 1991, and Figure 16 above).

Piecemeal examination of the model's errors revealed instances of each of the error-types described by Schwartz et al. (1998) as occurring in ADS. As shown in Table 6, the majority of errors were either omission or sequence errors. However, examples of object and gesture substitutions, action additions, tool omissions, and quality errors also occurred. As in ADS, the most frequent error type was omission. Such errors included omissions of both entire subtasks (usually the cream and/or sugar sequences) and fragments of subtasks. As the frequency of errors grew with increasing noise, the proportion of omission errors rose more rapidly than the proportions of other errors, as reported for ADS patients (Schwartz et al., 1998). Figure 19 shows the number of omission errors along with the number of sequence errors occurring at a range of noise levels. With increasing noise the number of omission errors grew steeply, while the number of sequence errors grew very little (compare Figure 17).

In summary, the behavior of the model reproduced several core characteristics of behavior in ADS: Deterioration in performance manifested as a gradually increasing fragmentation of sequential structure, a specific set of error-types occurred, and the proportion of omission errors increased with overall error rate.

Analyses

Fragmentation of Sequential Structure. A comparison of the results of the present simulation with those of Simulation 2 indicates a qualitative difference between the model's behavior under low vs. high levels of noise: At low noise levels, errors occurred primarily at the branch points between subtasks, while at higher levels of noise errors began to occur within subtask boundaries (see Figure 9). Despite the importance of this distinction, a close look at non-branch-point errors indicates that they involve precisely the same principles as the branch-point errors considered in the previous portion of the study. Once again, the model selects an incorrect action when noise causes its context representation to resemble a familiar pattern, connected with some other behavioral context, that is associated with that action. The only factor that distinguishes the situation away from branch-points from the branch-point case is that a greater degree of distortion is needed to produce the critical effect. At branch points, very small amounts of context distortion lead to errors, because of the close resemblance between the relevant temporal contexts. At non-branch-point steps, the contexts associated with different actions tend to be less similar to one

Table 5
Examples of the Model's Performance Under Increasing Noise

pick-up coffee-pack	pick-up coffee-pack	pick-up cup
pull-open coffee-pack	pull-open coffee-pack	sip
pour coffee-pack into cup	put-down coffee-pack	put-down cup
put-down coffee-pack	pick-up coffee-pack	pick-up carton
pick-up spoon	pour coffee-pack into cup	peel-open carton
stir cup	put-down coffee-pack	put-down carton
put-down spoon	pick-up spoon	pull-off sugar bowl lid
pull-off sugar bowl lid	stir cup	put-down lid
put-down lid	put-down spoon	pick-up spoon
pick-up spoon	pick-up sugar-pack	put-down spoon
stir cup	tear sugar-pack	pick-up coffee-pack
put-down spoon	pour sugar-pack into cup	put-down coffee-pack
pick-up carton	put-down sugar-pack	pick-up sugar bowl
peel-open carton	pick-up cup	put-down sugar bowl
pour carton into cup	put-down cup	pick-up coffee-pack
put-down carton	pull-off sugar bowl lid	pull-open coffee-pack
pick-up spoon	put-down lid	pour coffee-pack into cup
stir cup	pick-up spoon	put-down coffee-pack
put-down spoon	scoop sugar bowl with spoon	pick-up spoon
pick-up cup	put-down spoon	put-down spoon
sip cup	pick-up cup	pick-up cup
sip cup	sip cup	sip cup
say-done	sip cup	say-done
	say-done	

Note: Fixate actions are omitted. Left: This sequence skips from opening the sugar bowl directly to stirring, but is otherwise correct. Center: Among other errors, this sequence omits cream-adding and performs the sugar subtask twice (omitting steps in both cases). Right: Fragmented behavior under a high level of noise (variance 0.4).

Table 6
Examples of Individual Error-Types

Type	Example	Percentage
Omission	Sugar not added	77
Sequence		15
Anticipation	Pour cream without opening	
Perseveration	Add cream, add sugar, add cream again	
Reversal	Stir water then add grounds	
Other		8
Obj. substitution	Stir with coffee packet	
Gesture substitution	Pour gesture substituted for stir	
Tool omission	Pour sugar bowl into cup	
Action addition	Scoop sugar with, then put down, sugar bowl lid	
Quality	Pour cream four times in a row	

Note: Frequencies are based on a sample of 100 trials under noise with variance 0.2. Examples are drawn from sequences produced under a variety of noise levels.

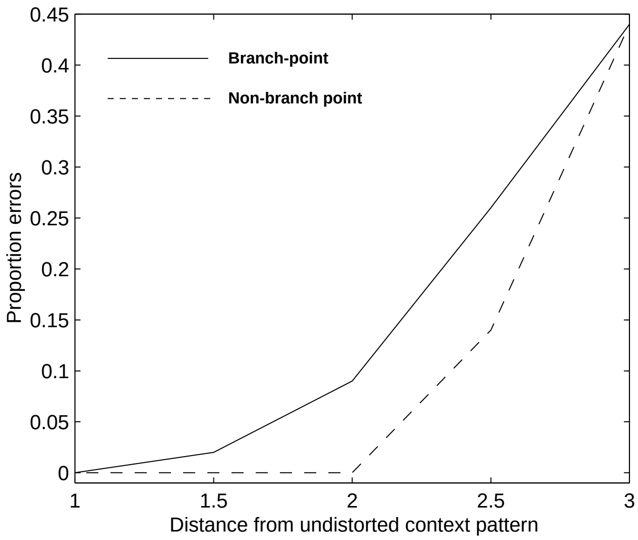

Figure 20. Greater distortion of context is needed to produce an incorrect action at a non-branch-point than a branch-point. The data shown were produced by adding progressively increasing random noise to the context representations normally arising at a branch-point step (the first step of the DRINK subtask in the sequence GROUNDS → SUGAR (BOWL) → CREAM → DRINK; solid line) and a non-branch-point step (the peel-open step in the sequence GROUNDS → SUGAR (BOWL) → CREAM; dashed line), and observing the outputs produced by the network. The x-axis shows the Cartesian distance of the distorted context representations from the noise-free pattern (binned in steps of 0.5). The y-axis shows the proportion of trials on which the incorrect action was selected.

another, meaning that a larger distortion of the model's context representation is needed to produce an error (Figure 20). Aside from this difference, which is one of degree, the basic factors that lead to errors at branch points and at non-branch points are identical.

Indeed, there is a sense in which *every* step in the sequences the model produces is a branch point. During training, the model has learned to associate every environmental input with a number of different actions, each linked to a different context. In some cases, two different actions may be associated with very similar contexts. As we have seen, this tends to be the case at the transitions between subtasks. However, even where the relevant contexts are more distinct, the network must use its representation of context to decide among possible actions. The model suggests that the dichotomy between branch points and non-branch points should be replaced by a view according to which each step in a routine falls somewhere on a continuum defined by the distinctiveness of the contexts associated with candidate actions. When the representation of temporal context is disrupted, errors occur first on steps located at one end of this spectrum, where different actions are associated with very similar contexts. With increasing disruption, errors begin to occur at points lying further and further along the continuum.

Gradedness of Fragmentation. As the data in Table 5 and Figure 18 make clear, the occurrence of an error did not throw the model completely 'off track.' Following an error, the model typically fell into behavior bearing some resemblance to the sequences presented during training. This tendency reflects the attractor dynamics that are characteristic of sequential networks (Gupta & Dell, 1999; Jordan, 1986a; Perlmutter, 1989). When placed in a state different from those occupied in executing sequences learned during training, recurrent models have a tendency to get drawn back into familiar lines of behavior over ensuing steps. The behavior of the present model under noise can be understood as reflecting a balance between this corrective tendency and the direct effects of noise; noise acts to knock the model out of learned patterns of behavior, and the model's attractor dynamics tend to draw it back into those patterns. With increasing noise, the former process increasingly dominates over the latter, and increasingly fragmented behavior is observed.

The Omission-Rate Effect. As overall error rate rose with increasing noise, the model produced an increasing proportion of omission errors. This finding is particularly significant, given the fact that a recent hierarchical model of ADS (Cooper & Shallice, 2000) did not reproduce the relationship between error-rate and the proportion of omission errors described by Schwartz et al. (1998). In the present model, there are two reasons for the increase in omissions. The first has to do with the fact that, with increasing noise, sequential behavior becomes more fragmented. Completion of the individual subtasks in coffee-making—as in most practical tasks—requires that a number of actions be executed in series. Naturally, as performance becomes more and more fragmented, the network develops an increasing tendency to get 'side-tracked' before reaching the end of these subsequences, and the goals of the relevant subtasks therefore tend to go unaccomplished.

The second factor underlying the omission effect is less obvious. It stems from a bias in action selection. Specifically, with increasing noise, the network shows an increasing bias toward selecting the actions pick-up and put-down and the fixate actions. Pour, tear, stir and the remaining actions are produced with diminishing frequency. This effect is reflected in Figure 21, which shows a step on which the correct response is peel-open. When the context representation is highly distorted, this action becomes rare in comparison with put-down, and the fixate actions.

Note that this effect cannot be reduced to a preference for the action most frequently performed in a given environmental context. Indeed, on the step illustrated in Figure 21, the peel-open action itself is the action most frequently associated with the current environmental input during training. Instead, the reason particular actions become predominant derives from the fact that, in the training set, they appear across a particularly wide range of contexts. The pick-up action appears in many contexts because it is a viable option whenever the grasp is empty. The put-down action appears in many because it is legal in all circumstances where the grasp is

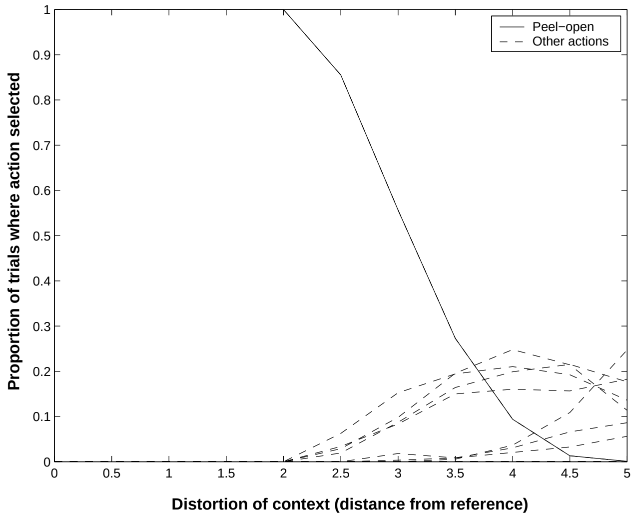

Figure 21. Analysis of action selection on the step in the coffee task where the correct action is peel-open, with the carton as the target object. As the distance between the distorted context representation and the reference pattern increases (x-axis), this action becomes decreasingly likely to be selected (solid line). At the same time, the context-general put-down and fixate actions (dashed) become more likely. The x-axis indicates centers of histogram bins. The y-axis indicates the proportion of trials within each bin on which a given action was selected. Actions indicated by dashed lines, in order of increasing activation at the far right of the figure, are fixate-teabag, fixate-coffee, fixate-sugarbowl, fixate-cup, fixate-spoon, put-down, and fixate-sugar.

occupied. The fixate actions are the most generic of all, since there is no situation where they are prohibited. Because the background training set contained examples that include as targets all actions that might plausibly be executed in a given environmental setting, these three actions appeared during training in an extremely wide variety of contexts. Actions more closely tied to specific environmental situations, such as tear, sip or peel-open, appeared in a significantly smaller variety of contexts.

To understand this correlation between the context-generality of actions and their robustness, consider what happens when the peel-open step shown in Figure 21 occurs with a highly degraded context representation. If sufficiently distorted, the context will not much resemble the one associated with the peel-open response. Instead, it is more likely to bear slight similarities to a wide range of contexts the network has encountered during training, contexts associated with both this and other external input. If a preponderance of these contexts are associated with a particular output, this output is likely to be selected. The network will therefore tend to produce outputs that are associated with a large range of contexts, i.e. put-down and fixate.

As we have noted, the model's bias toward context-general actions is one reason that it produces a rising proportion of omission errors with increasing noise. This is because, in the coffee task as in most others, successful completion of subtask goals depends on the execution of context specific actions. To the extent that action selection is biased against these, such goals will go unaccomplished, contributing to a disproportionate increase in errors of omission, relative to those of commission.

Beyond its contribution to the omission effect, the context-generality effect is important in that it gives rise to specific predictions about human behavior. We enumerate these, and assess their fit with some existing data, in the General Discussion.

General Discussion

Routine, sequential, object-directed action fills much of our daily lives. Despite the ubiquity of such action, its computational underpinnings remain incompletely understood. We have presented an account of how routine sequential activities are accomplished, based on the properties of recurrent connectionist networks, and supported this account with a series of simulations that exhibit several basic features of normal and impaired routine sequential behavior. The first simulation demonstrated the ability of the model to maintain a representation of temporal context capable of guiding behavior in circumstances where the environmental situation alone is ambiguous with respect to action, and capable of guiding performance in sequential tasks involving a flexible combination of subtasks. A slight degradation of this internal representation led to errors resembling everyday slips of action. In keeping with the empirical data, errors affected the ordering of subtasks more strongly than their internal structure. Interestingly, while errors tended to occur at the transitions between subtasks, the processes leading up to such errors were found to begin a number of steps earlier. In fact, noise applied midway through a subtask ultimately proved more disruptive to the model's performance than noise injected at the end of the subtask. This portion of the study also reproduced the reported effect of task frequency on error rate. Further degradation of the model's context representation led to a disruption of sequential structure within subtasks, yielding a pattern of behavior resembling that of ADS patients on a number of levels. Increasing degradation was accompanied by a graded increase in the frequency of independent actions; the errors produced by the model fell into the same categories as those produced by ADS patients; and as in ADS, a correlation was observed between overall error rate and the prevalence of omission errors. High levels of noise had an effect on the particular actions the network tended to select, resulting in a bias toward actions associated with a wide variety of objects and contexts.

In what follows, we consider the relationship between the present account and traditional models of routine sequence production, discuss some of the model's testable predictions, and consider its implications with respect to a number of key issues in the study of action.

Comparison with Hierarchical Models

The framework presented here differs from traditional accounts in at least two fundamental ways. First, the structure of the system's sequential behavior emerges from the functional properties of the processing system as a whole, rather than being linked in a direct fashion to discrete elements within the system's architecture. The system produces hierarchically organized behavior without relying on a structurally expressed schema hierarchy. Second, the representations that guide sequential behavior, and the detailed characteristics of the sequencing mechanism itself, develop through experience with relevant task domains rather than being built into the processing system *a priori*. These aspects of the present account allow it to avoid several basic problems associated with traditional models. As discussed in the Introduction, these problems include: 1) difficulty accounting for learning, 2) difficulty specifying a satisfactory sequencing mechanism, 3) difficulty coping with quasi-hierarchical domains and 4) difficulty accounting for certain kinds of error. In what follows, we revisit these four issues, considering them in the light of the present alternative framework.

Learning

The account we have presented here indicates how the knowledge structures underlying routine sequential behavior might develop through experience with specific tasks. When faced with the same issue, the traditional framework encounters difficulties. Specifically, it is not clear how individual computational elements within hierarchical models assume responsibility for particular segments of behavior, and how the appropriate links are formed from elements at one level to the levels above and below. The majority of work using hierarchical models circumvents the issue of learning, by building the needed structure into the processing system (e.g., Cooper & Shallice, 2000; Estes, 1972; MacKay, 1985; Rumelhart & Norman, 1982). Such extensive 'hand wiring' undermines the explanatory force of hierarchical models (Plaut & McClelland, 2000). In contrast to traditional accounts, the framework we have presented requires only very general assumptions about system architecture, leaving its detailed configuration to the learning process.

A small number of attempts have been made to provide computationally explicit accounts of learning in schema hierarchies. A representative proposal is provided by Houghton (1990; see also Grossberg, 1986). The procedure involves activating units representing the individual steps of the target sequence while a compound unit at the next level up in the hierarchy passes through an activation trajectory spanning the full duration of the sequence. During the sequence an associative learning rule is used to strengthen connections between the two levels. Importantly, in order to implement this learning scheme, it is necessary to assume a mechanism that detects the onset and termination of the sequences to be learned. For example, Houghton (1990) assumes a process that detects the boundaries of heard words, which supports learning of word pronunciations. However, empirical evidence suggests that, at least in some domains, reliable surface markers for event boundaries may not be readily available. As it happens, Houghton's (1990) chosen domain—speech—provides an excellent illustration of the point. Houghton suggests that word boundaries might be identified based on changes in sound intensity, but in naturalistic connected speech, word boundaries are associated with no such cues (see Morgan & Demuth, 1996). Indeed, there is evidence that the identification of event boundaries, at least in some contexts, depends on some prior knowledge of the sequential structure of the domain (see Avrahami & Kareev, 1994; Zacks & Tversky, 2001). For accounts like the one provided by Houghton (1990), this finding poses a chicken-and-egg conundrum: Acquisition of sequence knowledge depends on the ability to identify event boundaries, but the identification of event boundaries depends on sequence knowledge.

The framework we have proposed faces no such dilemma. To the extent that successful performance depends on information about task segmentation, the model derives this information from the statistical structure of the sequences encountered during learning. No surface indication of event boundaries is required. Indeed, it is significant that recurrent connectionist models have been used to account for the very processes that underlie the identification of event boundaries (Christiansen et al., 1998; Elman, 1990; Hanson & Hanson, 1996).

A further difference between the present framework and hierarchical accounts relates not to how learning is accomplished, but rather to what the system learns. In traditional accounts, acquisition of sequence knowledge involves instantiating a locally-represented schema, typically identified with a discrete node or processing unit. In the present account learning results in behavior that reflects schema-like knowledge, but without giving rise to any processing structure corresponding directly to a classical schema. Rather than implementing schema units, the system improves its performance by learning how to preserve and apply task-relevant contextual information. To put it strongly, while the notion of a schema may be useful in describing the system's behavior over time, the system itself contains no schemas at all. In this way, the account we have put forth here mirrors earlier work by Rumelhart et al. (1986, p. 21). As they put it,

> In the conventional story, schemata are stored in memory. Indeed, they are the very *content of memory*. In our case, *nothing stored corresponds very closely to* a schema. What is stored is a set of connection strengths which, when activated, have implicitly in them the ability to generate states that correspond to instantiated schemata. This difference is important—especially with regard to learning. There is no point at which it must be decided to create this or that schema. Learning simply proceeds by connection strength adjustment.... As the network is reorganized as a function of the structure of its inputs, it may come to respond in a more or less schema-like way.

Sequencing Mechanisms

The vast majority of work applying hierarchical models to action has been focused on the problem of specifying a basic sequencing mechanism. As discussed in the Introduction, even the most recent and sophisticated attempts to build sequentiality into hierarchical models continue to face rather fundamental difficulties, for example, problems in coping with cross-temporal contingencies, or in enforcing ordering among upper-level schemas. In the present account, such difficulties do not arise. Cross-temporal contingencies present no problem, since the model we have proposed, unlike hierarchical models, contains a sequencing mechanism capable of preserving specific information about previously selected actions. Nor is there a problem with ordering subtasks; the model provides a computationally explicit account of how sequencing is enforced at multiple levels of task structure. Indeed, at the most general level, the present theory differs from hierarchical accounts in that there is no need to build sequentiality into the underlying architecture. The fundamental characteristics of the system, in particular its connectivity, give rise to a general and remarkably flexible sequencing mechanism.

A key feature of this sequencing mechanism is the extent to which it is shaped by experience with specific tasks. This contrasts with hierarchical models, where sequencing mechanisms typically implement *a priori* assumptions about task structure. A pivotal example of this is the widely used mechanism of reflex inhibition. This implements a general assumption about the sequences the system will need to produce, namely that they will tend not to contain repeated elements. However, in reality, the frequency of repeats varies widely across behavioral domains (indeed, naturalistic behavior of the kind we have been considering is full of repeats). Furthermore, there is evidence that this variability has an impact on sequencing mechanisms; Vousden and Brown (1998) have observed that the frequency of repetition errors varies with the frequency of repetitions in the underlying task domain.

In contrast to hierarchical models, the model we have proposed builds in very few assumptions about sequential structure. The specifics of the sequencing mechanism are shaped by learning, with the result that they are closely adapted to the details of specific task domains. For example, with regard to repetitions, in domains where these are rare, the sequencing mechanism will develop a tendency to suppress completed actions (see Dell & O'Seaghda, 1994). In domains where repetitions are frequent, there will be a tendency to reactivate them. Indeed, such tendencies can be *item specific*. In our simulations of coffee-making, for example, the model learned to repeat the sip action, but not to repeat the pour action.

This last observation, concerning item-level repetition constraints, points to an empirically testable prediction. As just noted, Vousden and Brown (1998) reported different frequencies of repetition errors in domains with different base rates of repetition. By the model proposed here, this phenomenon should extend to the level of individual items within a single task domain: Repetition errors should more frequently involve items that tend to be repeated in the underlying domain, than items that are not repeated. Although Vousden and Brown (1998) suggested a way in which the mechanism of reflex inhibition might be elaborated to account for the data they reported, neither their account, nor any other hierarchical model of which we are aware, predicts such an item-level effect.

Dealing with Quasi-Hierarchical Structure

Another assumption that traditional models build in, concerning the structure of sequential tasks, is that such tasks will always be strictly hierarchical in structure. As a result, such models face difficulty in coping with situations where details of subtask performance depend on the larger task context, a circumstance very common in human behavior. In the present work, we have shown how a non-hierarchical system can overcome this problem, performing different versions of a subtask in different situations. The model's ability to produce context-sensitive behavior derives in part from the fact that it need not represent different levels of task structure disjunctively, as they are represented in hierarchical models. Instead, information pertaining to different levels of structure can overlap and interact in whatever way is needed to support task performance. Equally important is the model's use of distributed internal representations (Hinton, McClelland, & Rumelhart, 1986). Because such representations can capture graded similarity, they are able to encode different versions of a subtask in a way that acknowledges their overlap while still keeping them distinct.

Another way of viewing these aspects of the model is in terms of information-sharing among tasks. When a task resembles one that the system has already learned how to perform, the system will "re-use" representations associated with the familiar task in order to perform the new one. Schank and Abelson (1977; see also Schank, 1982) have argued that information-sharing of this kind is likely to be involved in the representations underlying human performance. As one example, they discuss the routines involved in eating in restaurants. While different types of restaurant—fancy restaurants, fast-food restaurants, cafeterias—call for different methods for obtaining a table, ordering, paying, etc., there is also a great deal of overlap in the behaviors they call for. Schank and Abelson argue that this overlap is reflected in the knowledge structures that guide restaurant behavior. Rather than there being a separate "script" for each different kind of restaurant, features common to all restaurants are represented once, with behaviors pertaining to specific kinds of restaurant built on top of this generic set of representations. The framework we have introduced here makes clear how this sort of representational scheme might be implemented, and how it might emerge from experience.

Traditional, schema-based accounts of action have sometimes implemented a form of information-sharing through the use of "slots" (e.g., Norman, 1981; Schank & Abelson, 1977). Here, schemas contain variables that can adopt several different specific values. While this approach allows a degree of information-sharing among related tasks, it entails a sharp distinction between tasks that share information structures and tasks that do not. This can lead to awkward dilemmas when the goal is to address domains where tasks have varying degrees of overlap. For example, it may seem reasonable to posit a single schema for spreading peanut butter and jelly, since these call for nearly identical actions. However, it is less clear whether this schema or some other should cover the weakly related tasks of spreading icing on a cake, sauerkraut on a hotdog, or wax on a car. In the present framework such dilemmas do not arise, since there is no discrete boundary between tasks that share representations and tasks that do not. The distributed representations the system employs allow it to implement a form of information-sharing that is well suited to the fact that tasks may overlap on many different dimensions and to widely varying degrees.

One important feature of this form of information-sharing is that it supports generalization. When faced with a novel situation, reasonable inferences can be made about appropriate actions based on the resemblance of the new situation to familiar ones. To return to the restaurant example from Schank and Abelson (1977), someone entering a Wendy's restaurant for the first time is likely to have a good sense of what to do, based on his or her prior familiarity with other fast-food restaurants. This sort of generalization falls naturally out of the processing framework we have considered here (for relevant simulations, based on the present model, see Botvinick & Plaut, in press).

Accounting for Pathologies of Action

Within hierarchical accounts of action, sequencing errors in both normal and apraxic performance have most frequently been understood as reflecting a weakening of the influence of explicit high-level schemas on lower-levels, or of lateral inhibition (Cooper & Shallice, 2000; MacKay, 1985; Schwartz et al., 1991; Shallice, 1988). The present account, in contrast, suggests that action errors result from the degradation of learned, distributed representations of temporal context.

As our simulations demonstrate, this proposal covers many of the same empirical phenomena addressed by the traditional account. However, it also offers some advantages. For example, the present framework appears to provide a more satisfactory account for the *graded* nature of action disorganization. This gradedness is evident in the data concerning recovery in ADS reported by Schwartz et al. (1991; see Figure 16), where the number of independent actions gradually fell over the weeks following initial evaluation. It is also evident across subjects, as in the population of patients reported by Schwartz et al. (1998; see Figure 17). Indeed, a graded continuum of action disorganization appears to connect the slips of normal subjects with the more severe errors of ADS patients (Schwartz et al., 1998). As we have noted, this spectrum appears to begin with disorganization primarily at the between-subtask level, with disorganization progressively infiltrating the within-subtask level as severity increases. It is not clear that this graded progression would fall out of the traditional, hierarchical account. Indeed, in the model of Cooper and Shallice (2000), gradually weakening top-down influence within a hierarchy of schemas resulted in abrupt, non-monotonic shifts in the degree and character of sequential disorganization.

As noted earlier, the Cooper and Shallice (2000) model also failed to capture two other aspects of human errors: the fact that slips of action sometimes involve repetitions of entire subtasks, and the fact that, in ADS, omission errors become increasingly predominant as overall error rate increases. In contrast, the present model reproduced both findings. Furthermore, the model provides a natural account for what Norman (1981) called "capture errors," lapses from one task into another following a series of actions that the two tasks share. Such errors follow naturally from the fact that in the present model, unlike hierarchical models, each action performed contributes to the system's representation of temporal context.

An influential idea concerning sequencing errors is that they reflect a failure of controlled as opposed to automatic processing. For example, Reason (1992) has suggested that naturalistic behavior consists of highly routine subsequences that can be executed without close attention, punctuated by decision points which require the actor to enter a special attentional mode. Norman (1981) makes a similar argument, calling these junctures "attentional checkpoints." Under this view, errors at the boundaries of subtasks are understood as reflecting a failure to engage the special attentional mechanisms responsible for guiding non-automatic behavior. In contrast to this account, the framework we have put forth involves no sharp distinction between automatic and controlled processing (see also Cohen, Dunbar, & McClelland, 1990), nor any sharp distinction between so-called decision points and other steps in processing. The phenomena that these constructs are meant to address—such as the tendency of slips to occur at the transitions between subtasks—emerge naturally from the system's basic sequencing mechanism.

At the same time that Reason has emphasized the distinction between controlled and automatic processing in explaining slips, he has also employed another idea that is much closer in spirit to the present account. Here, he suggests that errors may often result from "cognitive underspecification," where representations responsible for guiding behavior insufficiently specify the operations to be performed (Reason, 1992). The same theme appears in the work of Norman (1981), which discusses errors due to "insufficient description" (a term adopted from Norman & Bobrow, 1979). In essence, these accounts suggest that slips may occur due to internal representations that are in some way vague. The present account cashes out this intuition, making computationally explicit what this representational imprecision might involve. Underspecification emerges in the present account when the system's distributed representation of context is disrupted, producing a pattern that bears partial resemblances to familiar patterns linked to different behavioral contexts. In his arguments concerning cognitive underspecification, Reason (1992) has emphasized that "when cognitive operations are underspecified, they tend to default to contextually appropriate, high-frequency responses" (p. 71). The modeling work we have presented—in particular, the simulation addressing the relationship between task frequency and lapse errors—provides an mechanism-based explanation for why this is so.

Predictions

The present account gives rise to a number of testable predictions. In the following sections, we briefly review the predictions we have already had occasion to discuss, and add to them a number of further predictions.

Predictions Concerning Normal Performance

One distinctive aspect of our model is that it is sensitive to similarities among different temporal contexts that derive from the system's recent history. This sensitivity is most clearly evident in the model's errors, which can often be understood as involving a confusion between the present temporal context and some other familiar context to which it bears a resemblance. However, this property of the model is also manifest even when it does not commit errors, in biases that emerge in its sequencing behavior. To illustrate, let A, B, C and D stand for distinctive, multi-step action sequences. Let A' stand for a sequence resembling A. If the present model is trained, with equal frequency, on the four sequences A–B, C–D, A'–B, A'–D, it develops a tendency to produce A'–B more frequently than A'–D. The model predicts that this effect should be apparent in the behavior of human subjects learning to produce appropriately structured sequences in the laboratory. More generally, it should be possible to observe biases in the sequences produced by such subjects, based on an interaction between the objective structure of the target sequences and resemblances among the subsequences involved.

Predictions Concerning Slips of Action

Two predictions of the model have already been described: the differential effect of momentary distraction at or away from subtask boundaries, and item-specific effects on the frequency of repetition errors. A third prediction has to do with the tendency in the model for increasing noise to erode temporal structure beginning at the global level and extending to the internal structure of subsequences only later. The same pattern should be observed in normal subjects performing hierarchically structured sequential tasks under conditions of increasing distraction.

Predictions Concerning ADS

In Simulation 3, under high levels of noise, the model showed a bias toward context-general actions. If the model is correct, a similar bias should be observable in the behavior of ADS patients. Based on the model, context-general actions are assumed to include actions that respond to objects' basic mechanical affordances, including simply picking objects up and putting them down. Actions that relate to objects' conventional uses, or that require specific configurations of the environment are predicted to occur relatively infrequently. The model's predictions in this regard appear to receive some advance support from the informal observation of Schwartz et al. (1991) that ADS patients spend a great deal of time simply "toying" with objects. However, a more formal test of the prediction remains to be conducted.

A related prediction is that individuals with ADS should engage in frequent and prolonged periods of visual scanning during the performance of sequential tasks, since such action is itself highly context-general. While visual behavior in ADS has not been formally studied, prolonged periods of visual scanning have been observed in some patients (M. Schwartz and L. Buxbaum, personal communication).

If the model is valid, then it should also be possible to observe a bias toward context-general actions in normal subjects under conditions of distraction. Data that may be related to this prediction have been recently reported by Creem and Proffitt (2001). Here, subjects were asked to pick up individually presented familiar objects, each of which had a handle of one kind or another. In an initial test, Creem and Profit found that subjects lifted these objects using their handles, even when the handle was oriented away from them. However, under conditions of distraction, subjects tended to pick up objects without using their handles, grasping them elsewhere instead. While the task used in this experiment is not sequential in the usual sense, the results reported accord with the predictions of the present model by providing evidence that distraction can influence the affordances to which subjects respond.

Modeling Naturalistic Action: General Considerations

Our primary focus in the present work has been on sequencing phenomena. However, the framework we have presented also speaks to a number of other central issues in the domain of routine sequential action. One set of issues relates to the fact that much of everyday action is carried out upon objects. This raises the questions of how objects are selected as targets of action, and how the perception of objects may in turn impact the selection of actions. Another set of issues relates to the practical, goal-oriented nature of most routine sequential activity. This raises the question of how goals are represented, and how the relevant representations structure behavior. In what follows, we consider how these issues are dealt within the present computational framework.

Selection for Action

The present model adopts the view that object selection and action selection involve very similar procedures: Objects are first selected by performing perceptual actions, and then acted upon by executing manipulative actions, resulting in what Hayhoe (2000) has referred to as a "perception-action sequence." While this provides a framework for understanding a number of empirical findings relating to action with objects (as discussed in the Introduction), it does not answer the question of how the system determines which specific object to select at any given point in performance. According to the present account, this is accomplished by learned associations between particular environmental inputs and internal contexts, on the one hand, and specific perceptual actions on the other. Note that here, once again, the mechanisms underlying object selection are the same as those underlying the selection of manipulative actions. An interesting consequence of this is that disruptions of context representation affect not only the selection of overt actions, but also the selection of target objects. This provides a potential explanation for the fact that, in both everyday slips of action and in apraxia, sequencing errors tend to occur alongside errors in object selection. Indeed, in our simulations the majority of subtask omission and repetition errors began with errors in object selection (inappropriate perceptual actions).

Responding to Objects

Another key role for objects in the context of sequential routines is as *triggers* of action. A wide range of evidence indicates that action selection is directly influenced by perceived objects. The performance of subjects in laboratory tasks such as the Stroop (MacCleod, 1991) and flanker (Eriksen & Eriksen, 1974) tasks supports the idea that the perception of a visual stimulus leads to activation of strongly associated responses. Other data indicate that this phenomenon extends to complex stimuli familiar from everyday life (Riddoch, Humphreys, & Edwards, in press; Riddoch, Edwards, Humphreys, West, & Heafield, 1998; Tucker & Ellis, 1998).

The ability of objects to act as triggers of action is integral to the model we have proposed. During training, the model learns to associate perceptual inputs with particular actions. Presenting a given object to the fully trained model leads directly to activation of the associated responses. Of course, in the model as in human behavior, any given object or scene is likely to be associated with multiple responses. Action selection is thus a matter of selecting among the actions afforded by a given perceptual input. In the model, this is accomplished by combining information about the current stimulus with internally-maintained context information. The latter acts in effect as a filter on the external input, allowing it to trigger only the action appropriate to the present context.

This aspect of the model's function links it closely to several important models of cognitive control, which cast control as a top-down input biasing the system toward context-appropriate responses to incoming stimuli (e.g., Cohen et al., 1990; Norman & Shallice, 1986). An interesting aspect of such models, also shared by the present one, is that the context or control signal can specify a *class* of stimulus-response mappings, while allowing the system to select a specific response on the basis of perceptual inputs. Such an arrangement seems likely to be involved in routine actions on objects, where each execution of a given type of action must be fine-tuned to highly contingent aspects of the environment. For example, turning on the lights upon entering a room may involve flicking a switch, turning a dial, pushing a slider, pressing a button, etc. The framework we have presented here points to an account of how the action system might deal with such situations, using an internal representation of context to specify the broad class of action to be performed while allowing environmental inputs to determine the details of performance.

To this point, we have focused on the role of objects in triggering individual actions. However, in our model, object perception can also trigger entire behavioral sequences. Indeed, this occurs on each test run of the model. This ability of external inputs to trigger extended behavioral sequences fits well with human behavior. Duncan (1996) has emphasized that, because the environment is not entirely predictable, adaptive behavior requires the ability to enter new lines of behavior in reaction to environmental contingencies.

A final set of points concerning the role of objects in the present framework has to do with the way objects are represented by the processing system. The model's internal representations develop in order to facilitate the mapping from perceived objects to actions. One consequence of this is that objects that afford a similar set of actions become associated with similar internal representations.[10] This fits well with experimental data showing that human subjects tend to categorize objects on the basis of similarities among the actions they invite (Rosch, Mervis, Gray, Johnson, & Boyes-Braem, 1976). Another consequence of the learning process is that the system's internal representation of objects is more strongly influenced by perceptual features that predict afforded actions than by other features. This gives the processing system the potential to infer uses for objects, based on resemblances to other objects, a capacity that is present in human beings even at very young ages (Brown, 1990; Chen & Siegler, 2000).

It is significant that the model's learning about objects takes place entirely within the context of extended tasks. As a result, its internal representations integrate information about perceived objects with information about temporal context. One interesting implication of this is that degrading context information can impair the model's ability to represent the actions afforded by objects. In our simulations of ADS, for example, severe degradation to the model's representation of context biased against selection of context-specific actions, even when these were the actions most frequently associated with the presently fixated object. Context representation is equally dependent on object representation; impairing the model's ability to encode the features of objects should challenge its ability to maintain a useful representation of temporal context.

Bearing in mind this tight interdependence between object and context representation, it is interesting to consider an ongoing debate in the literature on ideational apraxia, a syndrome closely related to ADS. Some researchers have proposed that this form of apraxia primarily involves impaired knowledge concerning the use of objects (e.g., DeRenzi & Lucchelli, 1988). In contrast, others have suggested that

[10] Although the point can be demonstrated using the model presented here, earlier work already clearly shows how internal representations in connectionist networks are influenced by the similarity structure of output representations (see McClelland et al., 1995; Rogers & McClelland, in preparation).

the syndrome derives from impaired knowledge concerning the sequential structure of temporally-extended tasks (e.g., Lehmkuhl & Poeck, 1981). While our model does not directly address the data that inform this debate, it suggests that knowledge concerning object use and the knowledge that supports sequencing may be intertwined. Degradation to representations in either domain may manifest as impaired performance in the other, and damage at some levels of the processing system may impact both.

Representing Goals

It is frequently observed of human sequential behavior that it is organized around goals (e.g., Cooper & Shallice, 2000; Duncan, 1993; Fuster, 1989; Miller et al., 1960). Many observable aspects of behavior support this view: People often treat as interchangeable strategies that yield the same results; they monitor the outcome of their activities, evaluating whether they have brought about intended effects; and they compensate for unanticipated obstacles to action in a way that seems oriented toward the accomplishment of specific ends. In response to such behavior, some models of action incorporate special mechanisms for representing goals. Miller et al. (1960) posited TOTE units (test, operate, test, exit) as basic processing elements, one function of which is to compare the state of the environment with a goal state. Cooper and Shallice (2000) employed "goal nodes" as gates on activation flow between schemas at adjacent hierarchical levels (see Figure 2). Explicit goal representation also plays a central role in production system models of action (Anderson & Lebiere, 1998; Laird, Newell, & Rosenbloom, 1987; Newell, 1990).

While these efforts to address the goal-directedness of action highlight an important set of psychological and computational issues, they share at least two limitations. First, most existing computational accounts minimize the extent to which goals may be context dependent (a point emphasized by Agre, 1988). For example, one's goals in cleaning the house may vary widely depending on whether one is just tidying up or preparing for a visit from one's mother. Most existing models that depend on goal representations make no allowance for this context dependence. Second, there are many types of routine behavior for which it is not straightforward to identify discrete, explicit goals, for example taking a walk or playing the violin.

In contrast to traditional models, the model we have presented here does not rely on special goal representations to structure its behavior. The model enters into structured lines of activity not in response to the explicit setting of a goal, but because previous events have placed it in an internal state and environmental context that predisposes it toward those activities. Because the model's functioning does not depend on explicit goals, there is no problem in applying it to behaviors with non-obvious goals. Thus, understanding the activities of someone going outside for a walk does not require identification of the goals that prompted them to enter the relevant activities. On the present account, walk-taking could be triggered by feelings of restlessness, the thought of fresh air, the arrival of two o'clock (if one is in the habit of taking a walk then), or any other appropriate context.

However, while our model involves no special mechanisms for representing goals, it can produce behavior that appears goal-directed. For example, the model can learn to persist in a given action until a particular environmental criterion is met. This is apparent during the drinking sequence in the coffee-making task, where the model repeatedly selects the sip action until the cup is represented as empty. The model here implements something like the TOTE cycle described by Miller et al. (1960), continuing an activity until a particular goal is achieved.

Similarly, the model can learn to treat as interchangeable action sequences that address the same goal. For example, in our simulations, the model learned that the sugar-packet and sugar-bowl sequences could be used interchangeably. Thus, the model learned to associate the same contexts with both versions of the sugar subtask. While, in this case, the model was directly trained to perform two interchangeable sequences in the same set of contexts, systems of this sort can also infer sequence equivalence, interchanging equivalent sequences in a way that produces overall sequences the network has not observed during training (unpublished observations).

While the present model implements the view that organized action can occur without explicit goals, it seems clear that in some circumstances human action does involve instantiation of explicit goal representations. In principle, the present model could be elaborated to address such situations, possibly by including connections between the model's hidden units and a new group of units dedicated to representing desired states of the system or environment. Thus, the account we have presented is not intended to deny the existence or psychological importance of explicit goal representations. Nonetheless, it does treat skeptically the idea that such representations are fundamental to all routine sequential behavior (for related views, see Agre, 1988; Rogers & Griffin, submitted).

Challenges for the Present Account

Among the many questions raised by the work we have presented, an important one concerns the extent to which the present model is relevant to non-routine sequential behavior, for example that involved in problem solving, planning, error-detection and compensation, and coordination of multiple tasks. It is likely that in order to account for these aspects of behavior, additions to the model would be necessary. For example, the demands of planning appear to require some means of forecasting the outcome of one's actions, suggesting the addition of a forward model to the present architecture (Jordan & Rumelhart, 1992; see also Elsner & Hommel, 2001). It is possible that other elements, for example a mechanism supporting episodic memory, may also be necessary to support non-routine behavior (Cohen & O'Reilly, 1996; Ericsson & Kintsch, 1995). Nonetheless, we speculate that much of cognition and behavior, even in such domains as reasoning and problem-solving, may share a basic reliance on mechanisms of the sort illustrated in present model. It has been proposed that the cognitive operations involved in problem solving may themselves take the form of familiar, if very general-purpose, routines (see, e.g., Anderson & Lebiere, 1998; Chapman & Agre, 1987). Thus, at some level, problem solving itself may be a form of "routine" behavior. To the extent that this is the case, the mechanisms discussed in the current work may also be relevant to understanding the cognitive operations involved in problem solving and other non-routine behavior.

Another set of questions involves the relation between the behavioral phenomena addressed in the current work and other forms of routine sequential behavior. Models very similar to the one presented here have been proposed in work on language comprehension (Elman, 1991; McClelland et al., 1989) and production (including errors; Dell et al., 1993; Dell, Chang, & Griffin, 1999), raising the intriguing possibility that sequence production in language may rely on the same mechanisms as non-linguistic sequencing. In agreement with some others (e.g., Gupta & Dell, 1999), we suspect that a common set of computational principles and mechanisms underlie both linguistic and non-linguistic behavior. However, as will be clear from the foregoing discussion, the mechanisms in question adapt their behavior to the detailed structure of particular domains. The principles captured in the recurrent connectionist framework thus have the potential to play out in very different ways in the realms of language and non-linguistic action.

A further set of questions concerns the relation between the present model and the functional neuroanatomy underlying sequential action. Neuropsychological, neurophysiological and neuroimaging data point to a central role for several brain areas, including portions of frontal cortex (e.g., Fuster, 1995; Grafman, 1995), parietal cortex (e.g., DeRenzi & Lucchelli, 1988), the cerebellum and basal ganglia (see Hikosaka et al., 1999, for review). While the account we have presented does not attempt to delineate the division of labor among these brain areas, it does specify a set of computations that they may collaboratively support. Specifically, it suggests that this network of brain areas works together to maintain a representation of temporal context, integrating this with perceptual inputs in order to facilitate response selection. In this regard, it is interesting to note that features of the model we have presented bear a resemblance to features of existing computational models addressing the roles of prefrontal cortex (Cohen & Servan-Schreiber, 1992; Dominey, 1998) and basal ganglia (Berns & Sejnowski, 1998; Frank, Loughry, & O'Reilly, 2000).

From a biological perspective, there are also questions raised by the present model's use of the back-propagation learning algorithm. Although, in some studies, back-propagation learning has been shown to give rise to internal representations remarkably similar to those used by the brain (see, e.g., Zipser & Anderson, 1988), the algorithm does involve processes for which no biological correlate has yet been identified. Its use thus raises questions about biological plausibility. It is an open question whether the present theory could be implemented using a learning algorithm that is more consistent with current biological data (for example, the general recirculation algorithm proposed by O'Reilly, 1996). Of some relevance in this regard is the work of Dominey (e.g., Dominey, 1998; Dominey & Ramus, 2000), in which sequential networks are trained using a version of reinforcement learning. Aside from the learning algorithm it employs, the Dominey model differs from the one presented here in that its internal representations are fixed, rather than shaped through learning. This is concerning, given that many other modeling studies (e.g., Cleeremans, 1993; Elman, 1993, not to mention the present work) point to a key role for learned internal representations in complex sequential behavior. The question of how—and whether—sequential networks can be reconciled with constraints from neurobiology thus represents an important area for continuing work.

Another frequently voiced concern about back-propagation is its critical dependence on interleaved learning, without which the results of earlier training can be overwritten (McCloskey & Cohen, 1989). McClelland et al. (1995) have suggested that this may also be characteristic of a subset of human learning systems, and have provided an empirically motivated account for how the problem may be dealt with in the brain. The present account dovetails nicely with the McClelland et al. (1995) account, especially if the model is understood as simulating the consolidation of already acquired knowledge, as suggested in the Introduction. Nevertheless, it is important to acknowledge that the issues involved in this area are the subject of current debate (Page, 2000).

Conclusion

The domain of routine sequential activity raises a rich set of psychological issues, engaging the areas of perception, motor control, attention, and memory. A particularly central and poorly understood issue concerns how the cognitive system constructs and utilizes a representation of time-varying task context. The roughly hierarchical structure of many everyday tasks has led numerous theorists to the idea that the processing system itself assumes a hierarchical structure. We have proposed an alternative framework for understanding routine sequential action, which overcomes several basic limitations of the hierarchical approach. Here, task structure is represented not at the level of system architecture, but rather in the distributed representations the system uses in performing particular tasks. The system arrives at these internal representations through learning, leading to the emergence of sequencing mechanisms that are flexible, context-sensitive, and responsive to graded similarities among tasks.

Needless to say, the account we have presented abstracts over a great deal of important detail. Where possible, we have pointed to directions in which the model could be further developed in order to better capture detailed aspects of human behavior. Another limitation of the modeling efforts described here derives from the empirical data they address. Data concerning routine sequential behavior is scarce, and much of the available information is qualitative or anecdotal. In view of this, we consider it an important aspect of the present model that it makes several detailed and testable predictions concerning human sequential behavior.

Finally, one welcome aspect of existing research on routine sequential action is the degree to which relevant theories have been proposed in the form of explicit, implemented computational models. It is our hope that the work we have presented here will encourage the continuation of this trend.

Appendix: Coding of Model Performance

Counting of Independent Actions

In order to quantify the frequency of independent actions produced by the model, we adapted the coding scheme devised by Schwartz et al. (1991). The latter involves two steps: coding of individual actions, and a "bracketing" procedure according to which actions are grouped into subtask units centering on the achievement of a goal. Independent actions are ones that fall outside bracketed groupings.

Listing of Actions

Following the approach used by Schwartz et al. (1991), for each trial an initial listing was made of the sequence of actions (designated as "A1" in the Schwartz et al. work). Only overt actions were listed; fixate actions were omitted. In keeping with the details of the scheme of Schwartz and colleagues, pick-up actions were explicitly listed only if two or more cycles of processing elapsed prior to any action being performed with the object involved, and put-down actions were listed only if they occurred two or more cycles after the last action performed with the object. A single pick-up, put-down action was coded if an object was picked up and then put down within two time steps, with no action being performed using the object.

As in Schwartz et al. (1991), actions in the resulting listings were classifi ed according to the subtask to which they related: GROUNDS, SUGAR (BOWL), SUGAR (PACK), CREAM, STIR, or DRINK. A residual category for unassignable actions was included but rarely used.

Bracketing

Schwartz and colleagues used this term for a procedure for grouping A1's that lead up to the achievement of a "crux" action, an action that represents the completion of a given subtask. Independents are A1s that lie outside any bracketed group. The bracketing procedure we followed was based closely on the one described in Schwartz et al. (1991). Actions were grouped together if they derived from the same subtask and led up to a crux action (sipping, stirring, or pouring of coffee-grounds, cream, or sugar). Groups of actions were not grouped together if separated by a crux action– or more than one consecutive non-crux action–from another subtask. Following the bracketing procedure, each action falling outside of a bracketed group was counted as an independent.

Coding Scheme for Sequence and Omission Errors

To some extent, our categorization of the errors made by the model was made informally. However, a more explicit coding scheme was devised for the purposes of quantifying omission and sequence errors. Our approach was based on that adopted by Schwartz and colleagues in their empirical studies of ADS patients (e.g., Schwartz et al., 1991). This involved establishing a list of specifi c errors, based on a combination of preliminary observations of the network's performance and a consideration of the logical possibilities presented by the normative sequences. The lists used in generating the data presented in the paper were as follows:

Sequence errors
- Drinking prior to addition of all ingredients (if followed by additional ingredients)
- Repeated addition of any ingredient (with one or more intervening actions)
- Pouring cream prior to opening carton
- Scooping with spoon without opening sugar bowl
- Adding coffee only after adding other ingredients
- Stirring prior to adding any ingredients
- Repetition of stirring action (more than two consecutive stir actions)

Omission errors
- No coffee grounds added
- No sugar added
- No cream added
- Drinking omitted
- Ingredient added but not stirred in

Oila.
On Fri, 24 May 2002, David Plaut wrote:
¿ can you send me your revision.bbl fi le - thanks. ¿ d ¿

References

Agre, P., & Chapman, D. (1987). Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence* (p. 196-201).

Agre, P. E. (1988). *The dynamic structure of everyday life* (Tech. Rep. No. 1085). Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Allen, J., & Seidenberg, M. S. (1999). The emergence of grammaticality in connectionist networks. In B. MacWhinney (Ed.), *The emergence of language* (p. 115-151). Mahweh, NJ: Lawrence Erlbaum Associates.

Allport, A. A. (1987). Selection for action: Some behavioral and neurophysiological considerations of attention and action. In H. Heuer & A. F. Sanders (Eds.), *Perspectives on perception and action* (p. 395-419). Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review*, *94*, 192-210.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Hillsdale, NJ: Erlbaum.

Avrahami, J., & Kareev, Y. (1994). The emergence of events. *Cognition*, *53*, 239-261.

Baars, B. J. (1992). The many uses of error: 12 steps to a unified framework. In B. J. Baars (Ed.), *Experimental slips and human error: Exploring the architecture of volition* (p. 3-34). New York: Plenum Press.

Ballard, D. H., Hayhoe, M. M., Li, F., & Whitehead, S. D. (1992). Hand-eye coordination during sequential tasks. *Philosophical Transactions of the Royal Society, London (B), 337*, 331-339.

Ballard, D. H., Hayhoe, M. M., Pook, P. K., & Rao, R. P. N. (1997). Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences*, *20*(4), 723-767.

Berns, G. S., & Sejnowski, T. J. (1998). A computational model of how the basal ganglia produce sequences. *Journal of Cognitive Neuroscience*, *10*(1), 108-121.

Botvinick, M., & Plaut, D. C. (in press). Representing task context: Proposals based on a connectionist model of action. *Psychological Research*.

Brown, A. L. (1990). Domain-specific principles affect learning and transfer in children. *Cognitive Science*, *14*, 107-133.

Brown, G. D. A., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological Review*, *107*(1), 127-181.

Buxbaum, L. J., Schwartz, M. F., & Montgomery, M. W. (1998). Ideational apraxia and naturalistic action. *Cognitive Neuropsychology*, *15*, 617-643.

Chapman, D., & Agre, P. E. (1987). Abstract reasoning as emergent from concrete activity. In M. P. Georgeff & A. L. Lansky (Eds.), *Reasoning about actions and plans: Proceedings of the 1986 workshop* (p. 411-424). Los Altos, California: Morgan Kaufmann Publishers.

Chen, Z., & Siegler, R. S. (2000). Across the great divide: Bridging the gap between understanding of toddlers' and older childrens' thinking. *Monographs of the Society for Research in Child Development*, *65*(2).

Christiansen, M. H., Allen, J., & Seidenberg, M. S. (1998). Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes*, *13*, 221-268.

Christiansen, M. H., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, *23*(2), 157-205.

Cleeremans, A. (1993). *Mechanisms of implicit learning.* Cambridge, Massachusetts: MIT Press.

Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, *120*, 235-253.

Cohen, J. D., Dunbar, K., & McClelland, J. D. (1990). On the control of automatic processes: A parallel distributed processing account of the stroop effect. *Psychological Review*, *97*(3), 332-361.

Cohen, J. D., & O'Reilly, R. C. (1996). A preliminary theory of the interactions between prefrontal cortex and hippocampus that contribute to planning and prospective memory. In M. Bradimonte, G. O. Einstein, & M. A. McDaniel (Eds.), *Prospective memory: Theory and applications* (p. 267-295). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Cohen, J. D., & Servan-Schreiber, D. (1992). Context, cortex, and dopamine: A connectionist approach to behavior and biology in schizophrenia. *Psychological Review*, *99*(1), 45-77.

Cohen, N. J. (1984). Preserved learning capacity in amnesia: Evidence for multiple memory systems. In L. R. Squire & N. Butters (Eds.), *Neuropsychology of memory* (p. 83-103). Guilford Press.

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

Cottrell, G. W., & Plunkett, K. (1995). Acquiring the mapping from meanings to sounds. *Connection Science*, *6*, 379-412.

Creem, S. H., & Proffitt, D. R. (2001). Grasping objects by their handles: A necessary interaction between cognition and action. *Journal of Experimental Psychology: Human Perception and Performance*, *27*(1), 218-228.

Dell, G. S., Berger, L. K., & Svec, W. R. (1997). Language production and serial order: A functional analysis and a model. *Psychological Review*, *104*(1), 123-147.

Dell, G. S., Chang, F., & Griffin, Z. M. (1999). Connectionist models of language production: Lexical access and grammatical encoding. *Cognitive Science*, *23*(4), 517-542.

Dell, G. S., Juliano, C., & Govindjee, A. (1993). Structure and content in language production: A theory of frame constraints in phonological speech errors. *Cognitive Science*, *17*, 149-95.

Dell, G. S., & O'Seaghda, P. G. (1994). Inhibition in interactive activation models of linguistic selection and sequencing. In D. Dagenbach & T. H. Carr (Eds.), *Inhibitory processes in attention, memory, and language* (p. 409-453). San Diego: Academic Press.

DeRenzi, E., & Lucchelli, F. (1988). Ideational apraxia. *Brain*, *111*, 1173-1185.

Dominey, P. F. (1998). A shared system for learning serial and temporal structure of sensori-motor sequences? evidence from simulation and human experiments. *Cognitive Brain Research*, *6*, 163-172.

Dominey, P. F., & Ramus, F. (2000). Neural network processing of natural language: I. Sensitivity to serial, temporal, and abstract structure of language in the infant. *Language and Cognitive Processes*, *15*(1), 87-127.

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

Duncan, J. (1993). Selection of input and goal in control of behavior. In A. Baddeley & L. Weiskrantz (Eds.), *Attention: Selection, awareness, and control* (p. 53-71). New York: Clarendon.

Duncan, J. (1996). Cooperating brain systems in selective perception and action. In J. L. McClelland & T. Inui (Eds.), *Attention and performance 16: Information integration in perception and communication* (p. 549-578). Cambridge, MA: MIT Press.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179-211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, *7*, 195-225.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*, 71-99.

Elsner, B., & Hommel, B. (2001). Effect anticipation and action control. *Journal of Experimental Psychology: Human Perception and Performance*, *27*(1), 229-240.

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review*, *102*(2), 211-245.

Eriksen, B. W., & Eriksen, C. W. (1974). Effects of noise letters upon the identification of a target letter in a nonsearch task. *Perception and Psychophysics*, *16*, 143-149.

Estes, W. K. (1972). An associative basis for coding and organization in memory. In A. W. Melton & M. E. (Eds.), *Coding processes in human memory.* New York: Halsted Press.

Frank, M. J., Loughry, B., & O'Reilly, R. C. (2000). *Interactions between frontal cortex and basal ganglia in working memory: A computational model* (Tech. Rep. No. 00-01). Institute of Cognitive Science, University of Colorado, Boulder.

Fuster, J. (1995). Temporal processing. In J. Grafman, K. J. Holyoak, & F. Boller (Eds.), *Structure and functions of the human prefrontal cortex* (p. 173-181). New York: New York Academy of Sciences.

Fuster, J. M. (1989). *The prefrontal cortex.* New York: Raven Press.

Gaskell, M. G., Hare, M., & Marslen-Wilson, W. D. (1995). A connectionist model of phonological representation in speech perception. *Cognitive Science*, *19*, 407-439.

Gibson, F. P., Fichman, M., & Plaut, D. C. (1997). Learning in dynamic decision tasks: Computational model and empirical evidence. *Organizational Behavior and Human Decision Processes*, *71*, 1-35.

Grafman, J. (1995). Similarities and differences among current models of prefrontal cortical functions. In J. Grafman, K. J. Holyoak, & F. Boller (Eds.), *Structure and functions of the human prefrontal cortex* (p. 337-368). New York: New York Academy of Sciences.

Greenfield, P. M. (1984). A theory of the teacher in the learning activities of everyday life. In B. Rogoff & J. Lave (Eds.), *Everyday cognition: Its development in social context* (p. 117-138). Cambridge, MA: Harvard University Press.

Grossberg, S. (1986). The adaptive self-organization of serial order in behavior: Speech, language, and motor control. In E. C. Schwab & H. C. Nusbaum (Eds.), *Pattern recognition by humans and machines, vol. 1: Speech perception* (p. 187-294). New York: Academic Press.

Gupta, P., & Dell, G. S. (1999). The emergence of language from serial order and procedural memory. In B. MacWhinney (Ed.), *The emergence of language* (p. 447-481). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Hanson, C., & Hanson, S. J. (1996). Development of schemata during event parsing: Neissers perceptual cycle as a recurrent connectionist network. *Journal of Cognitive Neuroscience*, *8*(2), 119-134.

Hayhoe, M. (2000). Vision using routines: A functional account of vision. *Visual Cognition*, *7*, 43-64.

Henson, R. N. A., Norris, D. G., Page, M. P. A., & Baddeley, A. D. (1996). Unchained memory: Error patterns rule out chaining models of immediate serial recall. *Quarterly Journal of Experimental Psychology*, *49A*(1), 80-115.

Hikosaka, O., Nakahara, H., Rand, M. K., Sakai, K., Lu, X., Nakamura, K., Miyachi, S., & Doya, K. (1999). Parallel neural networks for learning sequential procedures. *Trends in Neurosciences*, *22*(12), 464-71.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence*, *40*, 185-234.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. volume 1: Foundations* (p. 77-109). Cambridge, Massachusetts: MIT Press.

Houghton, G. (1990). The problem of serial order: A neural network model of sequence learning and recall. In R. Dale, C. Mellish, & M. Zock (Eds.), *Current research in natural language generation.* San Diego, California: Academic Press.

Houghton, G., & Hartley, T. (1995). Parallel models of serial behavior: Lashley revisited. *Psyche*, *2*(25), 1-25.

Humphreys, G. W., & Forde, E. M. E. (1999). Disordered action schema and action disorganization syndrome. *Cognitive Neuropsychology*, *15*, 771-811.

Humphreys, G. W., Forde, E. M. E., & Francis, D. (2000). The organization of sequential actions. In S. Monsell & J. Driver (Eds.), *Attention and performance XVIII.* Cambridge, MA: MIT Press.

James, W. (1890). *The principles of psychology.* New York: Holt.

Jastrow, J. (1905). The lapses of consciousness. *Popular Science Monthly*, *67*, 481-502.

Jordan, M. I. (1986a). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the eighth annual conference of the cognitive science society* (p. 531-546). Englewood Cliffs, NJ: Erlbaum.

Jordan, M. I. (1986b). *Serial order: A parallel distributed processing approach* (Tech. Rep. No. 8604). University of California, San Diego, Institute for Cognitive Science.

Jordan, M. I., Flash, T., & Arnon, Y. (1994). A model of the learning of arm trajectories from spatial deviations. *Journal of Cognitive Neuroscience*, *6*(4), 359-376.

Jordan, M. I., & Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, *16*, 307-354.

Klatzky, R. L., Pellegrino, J., McCloskey, B. P., & Lederman, S. J. (1993). Cognitive representations of functional interactions with objects. *Memory and Cognition*, *21*(3), 294-303.

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

Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling.* Beverly Hills, CA: Sage University Series.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1-64.

Land, M. F., Mennie, N., & Rusted, J. (1998). Eye movements and the roles of vision in activities of daily living: Making a cup of tea. *Investigative Ophthalmology and Visual Science*, *39*(4), S457.

Lashley, K. S. (1951). The problem of serial order in behavior. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior: The Hixon symposium.* New York: John Wiley and Sons.

Lehmkuhl, G., & Poeck, K. (1981). A disturbance in the conceptual organization of actions in patients with ideational apraxia. *Cortex*, *17*, 153-158.

MacCleod, C. M. (1991). Half a century of research on the stroop effect: An integrative review. *Psychological Bulletin*, *109*, 163-203.

MacKay, D. G. (1985). A theory of the representation, organization, and timing of action, with implications for sequencing disorders. In E. A. Roy (Ed.), *Neuropsychological studies of apraxia* (p. 267-308). Amsterdam: Elsevier Science Publishers.

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

McCallum, A. K. (1996). *Reinforcement learning with selective perception and hidden state.* Unpublished doctoral dissertation, University of Rochester, Department of Computer Science.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why are there comlementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 419-457.

McClelland, J. L., St. John, M., & Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. *Language and Cognitive Processes*, *4*, 287-335.

McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, *24*, 109-165.

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

Morgan, J. L., & Demuth, K. (Eds.). (1996). *Signal to syntax: Bootstrapping from speech to grammar in early acquisition.* Mahwah, NJ.

Moss, H. E., Hare, M. L., Day, P., & Tyler, L. K. (1994). A distributed memory model of the associative boost in semantic priming. *Connection Science*, *6*(4), 413-427.

Munakata, Y., McClelland, J. L., & Siegler, R. S. (1997). Rethinking infant knowledge: Toward an adaptive process account of successes and failures in object permanence tasks. *Psychological Review*, *104*(4), 686-713.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: MIT Press.

Norman, D. A. (1981). Categorization of action slips. *Psychological Review*, *88*(1), 1-14.

Norman, D. A., & Bobrow, D. G. (1979). Descriptions: An intermediate stage in memory retrieval. *Cognitive Psychology*, *11*, 107-123.

Norman, D. A., & Shallice, T. (1986). Attention to action: Willed and automatic control of behavior. In R. J. Davidson, G. E. Schwartz, & D. Shapiro (Eds.), *Consciousness and self-regulation: Advances in research and theory.* New York: Plenum Press.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation*, *8*(5), 895-938.

Pacton, S., Perruchet, P., Fayol, M., & Cleeremans, A. (2001). Implicit learning out of the lab: The case of orthographic regularities. *Journal of Experimental Psychology: General*, *130*(3), 401-426.

Page, M. (2000). Connectionist modelling in psychology: A localist manifesto. *Behavioral and Brain Sciences*, *23*, 443-467.

Perlmutter, B. (1989). Learning state-space trajectories. *Neural Computation*, *1*(2), 263-269.

Plaut, D. C. (1999). A connectionist approach to word reading and acquired dyslexia: Extension to sequential processing. *Cognitive Science*, *23*(4), 543-568.

Plaut, D. C., & Kello, C. T. (1999). The emergence of phonology from the interplay of speech comprehension and production: A distributed connectionist approach. In B. MacWhinney (Ed.), *The emergence of language* (p. 381-415). Mahway, N.J.: Erlbaum.

Plaut, D. C., & McClelland, J. M. (2000). Stipulating vs. discovering representations. *Behavioral and Brain Sciences*, *23*.

Pylyshyn, Z. (1989). The role of location indexes in spatial perception: A sketch of the finst spatial-index model. *Cognition*, *32*, 65-97.

Reason, J. (1979). Actions not as planned: The price of automatization. In G. Underwood & R. Stevens (Eds.), *Aspects of consciousness, vol. 1: Psychological issues.* London: Academic Press.

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

Reason, J. T. (1984a). Absent-mindedness and cognitive control. In J. E. Harris & P. E. Morris (Eds.), *Everyday memory, actions, and absent-mindedness* (p. 113-132). London: Academic Press.

Reason, J. T. (1984b). Lapses of attention in everyday life. In R. Parasuraman & D. D. R. (Eds.), *Varieties of attention* (p. 515-549). London: Academic Press.

Reason, J. T. (1992). Cognitive underspecification: Its varieties and consequences. In B. J. Baars (Ed.), *Experimental slips and human error: Exploring the architecture of volition* (p. 71-91). New York: Plenum Press.

Riddoch, M. J., 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.

Riddoch, M. J., Humphreys, G. W., & Edwards, M. G. (in press). Visual affordances and object selection. In S. Monsell & J. Driver (Eds.), *Attention and performance XVIII.* Cambridge, MA: MIT Press.

Rogers, T. T., & Griffin, R. (submitted). *Goal attribution without goal representation: A connectionist approach to infants' early understanding of intential actions.*

Rogers, T. T., & McClelland, J. L. (in preparation). *Semantics without categorization: A parallel distributed processing approach to the acquisition and use of natural semantic knowledge.*

Rohde, D. L. T. (1990). *Lens: The light, efficient network simulator* (Tech. Rep. No. CMU-CS-99-164). Department of Computer Science, Carnegie Mellon Univeristy, Pittsburgh, PA.

Rohde, D. L. T. (2002). *A connectionist model of sentence comprehension and production.* Unpublished doctoral dissertation, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA. (Available as Technical Report CMU-CS-02-105.)

Rohde, D. L. T., & Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, *72*(1), 67-109.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, *8*, 382-439.

Roy, E. A. (1982). Action and performance. In A. W. Ellis (Ed.), *Normality and pathology in cognitive functions* (p. 265-298). London: Academic Press.

Rumelhart, D. A., & McClelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.

Rumelhart, D. E., Durbin, R., Golden, R., & Chauvin, Y. (1996). Backpropagation: The basic theory. In P. Smolensky, M. C. Mozer, & D. E. Rumelhart (Eds.), *Mathematical perspectives on neural networks* (p. 533-566). Hillsdale, NJ: Lawrence Erlbaum.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propogation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. volume 2: Psychological and biological models* (Vol. 1, p. 318-362). Cambridge, Massachusetts: MIT Press.

Rumelhart, D. E., & Norman, D. A. (1982). Simulating a skilled typist: A study of skilled cognitive-motor performance. *Cognitive Science*, *6*, 1-36.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986). Schemata and sequential thought processes in pdp models. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel dis-*

tributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models (p. 7-57). Cambridge, Massachusetts: MIT Press.

Sandson, J., & Albert, M. L. (1984). Varieties of perseveration. *Neuropsychologia*, *6*, 715-732.

Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people.* New York: Cambridge University Press.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Schneider, W., & Detweiler, M. (1988). The role of practice in dual-task performance: Toward workload modeling in a connectionist/control architecture. *Human Factors*, *30*, 539-566.

Schwartz, M. F. (1995). Re-examining the role of executive functions in routine action production. In J. Grafman, K. J. Holyoak, & F. Boller (Eds.), *Structure and functions of the human prefrontal cortex* (p. 321-335). New York: New York Academy of Sciences.

Schwartz, M. F., & Buxbaum, L. J. (1997). Naturalisic action. In K. M. Heilman (Ed.), *Apraxia: The neuropsychology of action* (p. 269-289). Hove, England: Psychology Press.

Schwartz, M. F., Mayer, N. H., Fitzpatrick, E. J., & Montgomery, M. W. (1993). Cognitive theory and the study of everyday action disorders after brain damage. *Journal of Head Trauma Rehabilitation*, *8*, 59-72.

Schwartz, M. F., Montgomery, M. W., Buxbaum, L. J., Lee, S. S., Carew, T. G., Coslett, H. B., Ferraro, M., & Fitzpatrick-DeSalme, E. J. (1998). Naturalistic action impairment in closed head injury. *Neuropsychology*, *12*(1), 13-28.

Schwartz, M. F., Montgomery, M. W., Fitzpatrick-DeSalme, E. J., Ochipa, C., Coslett, H. B., & Mayer, N. H. (1995). Analysis of a disorder of everyday action. *Cognitive Neuropsychology*, *8*, 863-892.

Schwartz, M. F., Reed, E. S., Montgomery, M. W., Palmer, C., & Mayer, N. H. (1991). The quantitative description of action disorganization after brain damage: A case study. *Cognitive Neuropsychology*, *8*(5), 381-414.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1988). *Encoding sequential structure in simple recurrent networks* (Tech. Rep. No. CMU-CS-183). Carnegie Mellon University, Department of Computer Science.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1991). Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning*, *7*, 161-193.

Shallice, T. (1988). *From neuropsychology to mental structure.* Cambridge: Cambridge University Press.

Stadler, M. A., & Frensch, P. A. (1998). *Handbook of implicit learning.* Thousand Oaks, CA: Sage Publications.

Tucker, M., & Ellis, R. (1998). On the relations between seen objects and components of potential actions. *Journal of Experimental Psychology: Human Perception and Performance*, *24*(3), 830-846.

Ullman, S. (1984). Visual routines. *Cognition*, *18*, 97-157.

Vousden, J. I., & Brown, G. D. A. (1998). To repeat or not to repeat: The time course of response suppression in sequential behavior. In J. A. Bullinaria, D. W. Glasspool, & G. Houghton (Eds.), *Proceedings of the fourth neural computation and psychology workshop* (p. 301-315). London: Springer-Verlag.

Vousden, J. I., Brown, G. D. A., & Harley, T. A. (2000). Serial control of phonology in speech production: A hierarchical model. *Cognitive Psychology*, *41*(2), 101-175.

Wall, L., Christiansen, T., & Orwant, J. (2000). *Programming perl (3rd edition).* Cambridge, Massachusetts: O'Reilly and Associates.

Whitehead, S. D., & Ballard, D. H. (1990). Active perception and reinforcement learning. *Neural Computation*, *2*, 409-419.

Whitehead, S. D., & Lin, L. (1995). Reinforcement learning of non-markov decision processes. *Artificial Intelligence*, *73*, 271-305.

Williams, R. J., & Zipser, D. (1995). Gradient-based learning algorithms for recurrent networks and their computational complexity. In Y. Chauvin & D. E. Rumelhart (Eds.), *Back-propagation: Theory, architectures and applications* (p. 433-486). Hillsdale, NJ: Erlbaum.

Zacks, J., & Tversky, B. (2001). Event structure in perception and conception. *Psychological Bulletin*, *127*, 3-21.

Zipser, D., & Anderson, R. A. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, *331*, 679-684.