Psychological Research (2002) 66: 298–311
DOI 10.1007/s00426-002-0103-8

ORIGINAL ARTICLE

Matthew Botvinick · David C. Plaut

Representing task context: proposals based on a connectionist model of action

Accepted: 18 March 2002 / Published online: 7 November 2002

Abstract Representations of task context play a crucial role in shaping human behavior. While the nature of these representations remains poorly understood, existing theories share a number of basic assumptions. One of these is that task representations are discrete, independent, and non-overlapping. We present here an alternative view, according to which task representations are instead viewed as graded, distributed patterns occupying a shared, continuous representational space. In recent work, we have implemented this view in a computational model of routine sequential action. In the present article, we focus specifically on this model's implications for understanding task representation, considering the implications of the account for two influential concepts: (1) cognitive underspecification, the idea that task representations may be imprecise or vague, especially in contexts where errors occur, and (2) information-sharing, the idea that closely related operations rely on common sets of internal representations.

Introduction

It seems reasonable to view human action as typically reflecting two primary influences: a pattern of input from the environment, and an internal representation of the behavioral context. The flexibility of human behavior is such that the first of these influences, perceptual input, is almost never sufficient fully to determine action selection. Whether the case involves an experimental participant in front of the computer screen, or a person at home looking through the silverware drawer, action selection requires that perceptual inputs be combined with information about the task at hand.

How is task context represented internally? Certainly, important progress toward answering this question has been made through experimental work focusing on task-switching, dual-task performance, and related phenomena. However, the assumptions made about task representations in such work tend to remain somewhat implicit, manifesting only in the use of terms such as "activation", "priming", or "decay". For explicit proposals concerning the nature of task representations, one must turn to the computational modeling literature. Interestingly, despite the diversity of views expressed in this literature, most accounts seem to share some basic assumptions about task representations, assumptions that also appear to inform discussions of empirical studies.

In this article, we consider one of the most pervasive assumptions about task representations, and present a theoretical alternative. As explained in the next section, the majority of available models portray task representations as discrete, independent entities bearing little structural relation to one another. In recent work, we have put forth a model of action that portrays task representations in quite a different way. Here, task representations are viewed as occupying a common representational space, sharing graded similarity relations along multiple dimensions. This view turns out to have far-reaching implications for understanding behavior in complex domains, casting a new light on some familiar theoretical constructs.

The traditional approach

Some traditional assumptions about task representation can be illustrated using the computational model of

M. Botvinick (✉)
Department of Psychiatry, University of Pensylvania
School of Medicine and Center for Cognitive Neuroscience,
University of Pennsylvania
E-mail: mmb@mail.med.upenn.edu
Tel.: +1-215/898-2960
Fax: +1-215/898-1982

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

sequential action recently proposed by Cooper and Shallice (2000). This model, which addresses the everyday task of coffee-making, takes the form of a hierarchical tree structure made up of "schema nodes." At the base of the hierarchy, each node represents a comparatively simple action (e.g. pouring or stirring). As one rises through the hierarchy, nodes relate to larger and larger segments of the overall task. There is a subtask level, where units represent subroutines such as adding cream or sugar, and, at the top of the hierarchy, a task level where a single schema unit is used to represent the overall task of coffee-making.

The "coffee-making" unit at the summit of the Cooper and Shallice (2000) model is emblematic of one pervasive assumption running through many models of task representation. The assumption is that task representations are *discrete, non-overlapping, and independent*. In order to make this point clear, imagine that the Cooper and Shallice (2000) model were to be expanded so as to address not only coffee-making but also the related task of tea-making. This would involve adding a new "tea-making" unit to the model at the task level, completely independent of the coffee-making unit.

Of course, the fact that tasks are represented independently does not mean that their representations are *functionally* independent. Thus, for instance, the coffee- and tea-making units in the foregoing example might share connections that place them in competition. In addition, the tea unit might send activation to some of the same subtask units as the coffee-making unit. Later in this article, we will consider the implications of these functional connections between task representations. However, since the central points we wish to make pertain to the properties of individual task representations, we will, initially, focus on such representations as the unit of analysis.

A view of task representations as discrete, independent, non-overlapping entities is shared by many computational models of human action sequencing. With a few notable exceptions, such models tend to fall into one of two traditions. The first of these involves the use of hierarchies of localist units, as seen in the Cooper and Shallice (2000) model and earlier in models proposed by Miller, Galanter & Pribram (1960), Estes (1972), Mackay (1987), Rumelhart & Norman (1982), and Houghton (1990), among others. The second tradition involves the use of production system architectures (e.g. Anderson & Lebiere, 1998). Here, task representation relies crucially on goal representations, which (in ACT-R and some other schemes) are managed by a "stack" mechanism. Like the task representations in hierarchical models, and like the productions they trigger, these goals are represented in a discrete, independent, non-overlapping fashion.

An alternative account

In the present article, we wish to propose an alternative view of task representations. Here, rather than standing independently of one another, task representations overlap structurally, sharing graded, multidimensional similarity relations. By this view, one can think of task representations as occupying a shared representational space, with representations for interrelated tasks lying close to one another, in much the way representations of objects or concepts are sometimes thought of as occupying a semantic space.

We have recently implemented this view in a computational model of action. As discussed in a separate report (Botvinick & Plaut, 2002), this model has been applied to a variety of empirical data pertaining to the performance of routine sequential tasks. Botvinick & Plaut compare the model to traditional accounts of sequential behavior, pointing to several important advantages. We will briefly summarize some of the relevant points below. However, the more focal objective of the present article is to consider the implications of the model for concepts of task representation.

Some of the model's key implications in this regard can be brought out most clearly by considering how it relates to some preexisting theoretical proposals concerning task representation. In what follows, we focus specifically on two such proposals. The first, coming out of work on action slips (e.g. Reason, 1990), is the idea that task representations can be imprecise or underspecified. The second idea, from work on complex naturalistic behavior by Schank and others (Schank, 1982; Schank & Abelson, 1977), is that individual task representations may be "shared" by multiple, structurally interrelated activities. Taken as claims about the nature of individual task representations, these ideas appear difficult to square with accounts in which tasks are represented in an independent and non-overlapping fashion. In contrast, underspecification and information-sharing arise as inherent properties of task representation in the model to which we now turn.

A model of routine sequential action

The model of routine sequential action we have put forth in recent work builds on earlier research using recurrent connectionist networks. The details of our simulations are described by Botvinick & Plaut (2002). Here we provide an overview, concentrating on elements most relevant to the issue of task representation.

Model architecture and task domain

The structure of the model is shown in Fig. 1. Like all connectionist models, it comprises simple processing units, each with a scalar activation value. These excite or inhibit one another through adjustable, weighted connections. In the current model, units are organized into three groups. A group of input units serves to represent the perceptual features of objects in the environment. These units connect to an internal or "hidden" group,

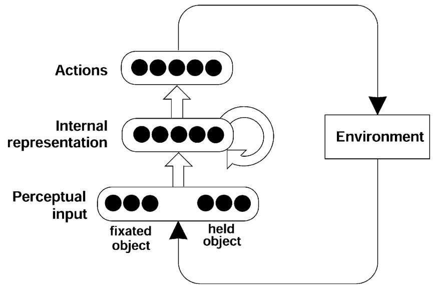

Fig. 1 Schematic representation of the model reported by Botvinick & Plaut (2002). *Arrows* denote connections running from each unit in the sending group to each unit in the receiving group. In the actual implementation, the input layer contained 44 units, the hidden layer 50, and the output layer 20

which itself connects to an output group whose units represent simple actions (e.g. "pick-up," "pour", or "locate-spoon"). In order to capture the fact that actions affect perceptual inputs, the model communicates with a simulated environment, which updates inputs to the network contingent on selected actions.

A crucial feature of the model is that there are reciprocal connections between each pair of units in its internal layer. The presence of these "recurrent" connections means that activation can flow over circuits within the network, allowing information to be preserved and transformed over multiple steps of processing. It also has important implications for the role of the model's internal units. Given their overall pattern of connectivity, these units play two roles. First, they serve as an intermediate stage in the stimulus-response mapping performed on each processing step. Second, because they carry all of the information that will be conducted over the network's recurrent connections, and thus all of the information that will be carried over to the next time-step, they are responsible for carrying the model's representation of temporal context. Much of the ensuing discussion will focus on the model's internal units as they fulfill the second of these two roles.

A number of studies have demonstrated the ability of recurrent networks to address aspects of human behavior in the domains of language (e.g. Elman, 1990) and implicit learning (e.g. Cleeremans, 1993). Our simulations investigated whether similar computational principles could be used to account for human behavior in everyday, goal-oriented tasks involving the manipulation of objects. In order to facilitate comparison with the recent hierarchical model of Cooper & Shallice (2000), the task modeled was that of making a cup of instant coffee. Our implementation of the task is shown schematically in Fig. 2 (top). It comprises four subtasks, each containing between five and 11 actions: (1) adding coffee-grounds, (2) adding cream, (3) adding sugar (by one of two methods), and (4) drinking. For reasons that will become clear in later discussion, the training corpus also contained a second task, tea-making. The model was trained to perform these tasks using a version of the back-propagation learning algorithm (Williams & Zipser, 1995). Training was analogous to observing and attempting to predict the sequence of actions of a skilled individual repeatedly carrying out specific versions of each task. Testing involved successively presenting the trained model with perceptual input and using its generated action to modify the environment (and, hence, the model's subsequent perceptual input).

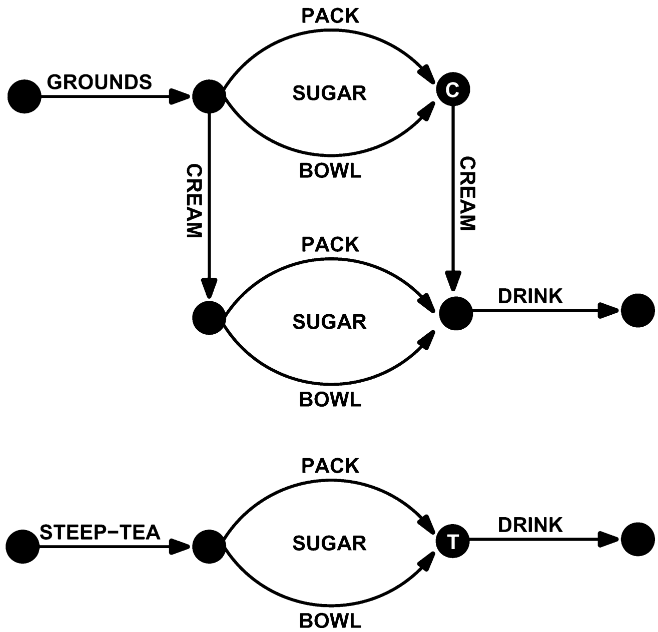

Fig. 2 Schematic representation of the tasks included in the training set. Each arrow corresponds to a multi-step sequence (range 5–11 steps)

Overview of simulation results

Simulations were designed to address behavioral data from three domains: (1) normal, error-free performance in hierarchically structured tasks, (2) everyday "slips of action," and (3) action disorganization syndrome (ADS), a variety of apraxia involving impairment in the performance of everyday tasks. In our simulations of normal performance, we asked simply whether the model could learn to perform the target tasks. Some action researchers have expressed doubt concerning the ability of recurrent networks to deal with tasks that are hierarchically structured, that is, tasks made up of subtasks and actions that also appear as part of other tasks (see, e.g. Houghton & Hartley, 1995). Consistent with earlier studies applying recurrent networks in hierarchical domains, the model proved quite capable of learning the target sequences, and producing them autonomously following training.

Our simulations of action slips and ADS were based on the assumption that both stem from disruptions to representations of temporal or task context. In our model, as noted above, such context information is carried by the hidden units. With this in mind, context information was degraded by randomly perturbing the activation values in the hidden layer on each cycle of processing. When this was done mildly, the model produced errors resembling human slips of action. In line with empirical observations concerning slips (Norman, 1981; Reason, 1990), the model made errors at decision points, behavioral "forks in the road" where the actions just completed bear associations with multiple lines of subsequent behavior. Also like typical human slips, the model's errors took the form of subtask sequences performed correctly but in the wrong context. The model's errors fell into the same categories as human slips: omissions, repetitions, and lapses from one task into another. With increasingly severe disruption to the model's context representations, the model's behavior became gradually more fragmented, coming to resemble the performance of ADS patients as characterized in recent empirical studies (e.g. Humphreys & Forde, 1999; Schwartz et al., 1998).

Botvinick & Plaut (2002) point to a number of apparent advantages of the model over traditional accounts of routine sequential action. Some of these pertain to the model's ability to capture particular behavioral phenomena. Specifically, the model produced at least one type of error (recurrent perseveration) not observed in the simulations of Cooper and Shallice (2000); it reproduced a correlation between error rate and the distribution of error types reported by Schwartz et al. (1998), another effect not captured by Cooper and Shallice (2000); and, again unlike that earlier study, the Botvinick & Plaut (2002) model displayed a smooth variation in behavioral fragmentation with damage, a feature of ADS. Botvinick & Plaut (2002) also discuss several other advantages of the model over traditional accounts, including its reliance on learning instead of extensive "hand wiring," its avoidance of the inflexible, ad hoc sequencing mechanisms typically incorporated into traditional models, and its relative strength in dealing with context-sensitive behavior. However, rather than reiterating this critique, our goal here is to examine the model's account of task representation. We turn now to this issue.

Graded, distributed representations of task context

Whether the model is used to simulate normal performance or errors, its behavior is linked directly to the patterns of activation over the units in its internal layer. As noted above, these units play two roles. Because they lie between input and output layers, they are responsible for facilitating the stimulus-response mapping being performed on each time-step. Second, because, via their recurrent connections, the internal units transmit information from one time-step to the next, they also must serve to represent the current behavioral context. In this sense, the patterns of activation arising in the model's internal layer play the role that is played, in traditional models, by task and subtask nodes; on each time-step, the information carried in this layer is integrated with information about external inputs in order to determine the context-appropriate action.

Note that every unit in the hidden layer participates in each context representation. Unlike hierarchical models of action, which use single units to represent entire task contexts, the present model employs *distributed* representations (see Hinton, McClelland, & Rumelhart, 1986); information is represented by an entire population of processing units, within which each unit participates in representing a variety of contexts.

In order to understand the implications of the model's way of representing context, it is useful to adopt a spatial metaphor. The model's internal layer contains 50 units, each of which carries an activation between zero and one. If these activations are thought of as spatial coordinates, then each pattern of activation (context representation) can be thought of as specifying a point in a 50-dimensional representational space. As the model steps through an action sequence, the successive patterns in its internal layer can be thought of as tracing out a trajectory in this space. Although it is impossible to visualize such trajectories in their original 50 dimensions, one can gain a sense of them using the technique of multi-dimensional scaling (MDS). This allows trajectories in high-dimensional space to be represented in two dimensions, while preserving as much information about the original pattern as possible (see Kruskal & Wish, 1978). Perhaps the most important aspect of the results yielded by MDS is that they carry information about the *similarities* among the model's internal representations. Such information is conveyed through the proximities of points within the resulting diagram. To a first approximation, points located near to one another correspond to patterns of activation that are similar to one another, while points located distant from one another correspond to more dissimilar patterns of activation.

An example of the model's internal representations, visualized with MDS, is shown in Fig. 3. The plot shows two trajectories, both representing the sequence of internal states produced by the model as it stepped through the eleven actions of the sugar-adding subtask (the solid line shows the patterns produced when performing the subtask in the context of coffee-making, the dashed line when performing it during tea-making). Together, these trajectories provide a particularly clear illustration of the importance of graded similarity relations in the model's representation of task context. The first thing to note is that the two trajectories are similar in shape. This indicates that the series of internal representations the model uses when adding sugar to coffee are similar to those it uses when adding sugar to tea, an arrangement that makes sense since sugar-adding involves the same sequence of actions regardless of the overall task context. Note, however, the two trajectories are not precisely identical. The minor differences between the two reflect the difference in overall task context; the model's internal representations on each step differ slightly according to whether it is coffee- or

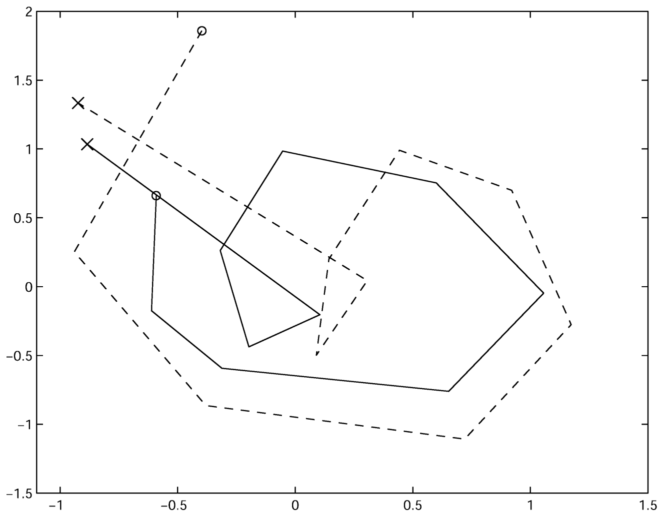

Fig. 3 Multidimensional scaling analysis of internal representations from the model. Each point corresponds to a 50-dimensional pattern of activation across the network's hidden units. Both traces are based on patterns arising during performance of the sugar-packet subtask. The *solid trajectory* shows patterns arising when the sequence was performed as part of coffee-making, the *dashed trace* when it was performed as part of tea-making

tea-making that is being performed. As earlier studies of recurrent networks (e.g. Servan-Schreiber, Cleeremans, & McClelland, 1991) have expressed it, the network "shades" its internal representations to reflect differences in context.

This first example illustrates a key point: Because the model's internal representations occupy a continuous, multidimensional representational space, they can simultaneously capture both the similarities and the differences among behavioral contexts. In the next two sections, we examine the implications of this point for understanding the representation of task context, using the notions of underspecification and information-sharing as points of departure.

Underspecification

An interesting idea that has surfaced periodically in work on action slips is that such errors might result when task representations are in some sense degraded or vague. The notion that task representations might vary in their preciseness was suggested early on by Norman & Bobrow (1979). Soon after, Norman (1981) linked the idea to error-commission, writing that "Some slips of selection occur ... when the description of the desired act is insufficient" (p. 7). This notion has been emphasized and developed further in the work of Reason (e.g. Reason, 1990, 1992), who coined for it the term "underspecification."

The idea of underspecification has an appealing intuitiveness. However, discussions of it have always remained rather impressionistic. This is perhaps because the idea that task representations can vary in their preciseness lies at odds with other pervasive assumptions about task representations. As we have seen, traditional models of action typically portray task representations as discrete, unitary entities. In what sense can a "schema node" like the ones posited by Cooper and Shallice (2000) become vague? To be sure, models employing such task representations can assume ambiguous states where more than one schema node is active (a point whose implications we will consider in a later section). However, this amounts to something different from underspecification, which suggests that vagueness (or precision) is a property of individual representations, considered in their own right.

Underspecification, in this sense, becomes easier to envision if task representations are thought of as points in a shared and continuous representational space. Under this view, one can think of the space of task representations as populated by certain prototype representations, points in the space corresponding to familiar task contexts. A novel situation or a disturbance in the system's functioning may then give rise to a representation occupying a location midway among several of these reference representations, rendering the representation ambiguous. In the next section, we detail this account and its implications for understanding action slips by considering a specific error committed by the recurrent network model.

Simulating an action slip

In our simulations, slips of action were elicited by degrading the model's internal representations, adding random noise to each unit's activation. This led to a variety of error types, and an overall pattern of behavior resembling human slips as characterized by Reason (1990) and others. The full range of findings is described in Botvinick & Plaut (2002). In order to focus discussion on representational issues, we will focus here on one particular error committed by the model: adding cream to tea.

As shown in Fig. 2, our implementation of the tea task overlapped with the coffee task in that both involved sugar-adding and drinking subtasks. An important difference was that coffee-making included a cream-adding subtask, whereas tea-making did not. This arrangement was useful when it came to asking whether the model's errors included lapses from one task into another, a form of error often observed in human behavior. If the model began by steeping tea and adding sugar, but then executed the cream-adding sequence, this could be clearly interpreted as a lapse from tea-making into coffee-making. This error did in fact occur regularly when the model's internal representations were degraded.

The cream-into-tea error, like any lapse error, has the issue of task representation at its base. The error can be understood as involving a confusion between two task contexts. At the point the error occurs, the model is at the point in the tea task labeled with a *T* in Fig. 2.

However, the model behaves instead as if at the point labeled with a *C*. How does this confusion occur? In approaching this question, it is useful to consider how the model avoids the cream-into-tea error in normal operation. Here, of course, it relies on its internal representation of task context. Through learning, the model arrives at specific patterns of activation to represent the contexts labeled in Fig. 2. In fact, these patterns, referred to from here on as the tea and coffee contexts, are represented in Fig. 3; they correspond to the final points in the two trajectories, the internal representations arising as the model completes the sugar-adding subtask.

Figure 4 shows schematically how the model comes to mistake the tea context for the coffee context. The two reference patterns are represented, once again, as points in a shared representational space. The crooked arrow illustrates the effect of degrading the model's internal representation of context. Perturbing the activation values in the model's hidden layer corresponds to randomly displacing the model's context representation in representational space. In some instances, this random displacement may carry the model's context representation toward the point ordinarily used to represent the coffee context. If the distorted representation falls near enough to this reference point, the model will behave as if it is performing the coffee task rather than the tea task, thus committing the cream-into-tea error.

Figure 5 shows relevant data from the simulations themselves. Each bar in this plot indicates the distance between the model's distorted internal representation and one reference pattern. The light gray bars show the distance between the distorted pattern and the tea-context reference, the darker bars the distance between the same pattern and the coffee reference. The left half of the diagram shows data from trials where, despite the presence of noise, the model did not commit the cream-into-tea error. Here, although the model's internal representations have been displaced from the position used to indicate the tea context, they remain closer to this reference point than to the coffee-context representation. On trials where the cream-into-tea error did occur, the situation is reversed (Fig. 5, right). Here, noise has carried the model's internal representations farther away from the tea-making reference, into the vicinity of the coffee-making representation. In effect, representational degradation causes the model to "forget" that it is making tea, causing it to lapse out of this task and into another.

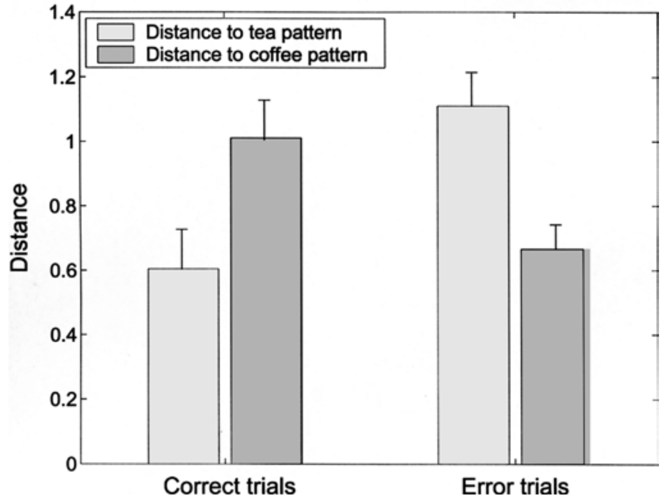

Fig. 5 Simulation data, showing distances of degraded internal representations from reference representations. All data were collected on the final step of sugar-adding. Reference patterns were drawn from simulations of the tea and coffee tasks, run without noise. Distorted representations are from simulations of the tea task, using gaussian noise with variance 0.1 (for detailed methods, see Botvinick & Plaut, 2002). *Bars* show the average distance of distorted representations from the tea (pale gray) and coffee (dark gray) reference patterns. Data are based on a sample of ten correct and ten error trials. Error bars show standard error

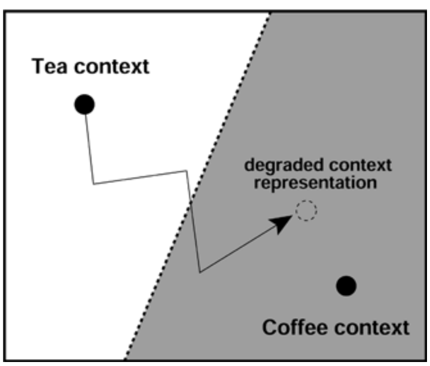

Fig. 4 Adding noise to the model's internal representations has the effect of displacing them within representational space. In some instances, the resulting distortion can lead the internal representations to resemble patterns the model uses to represent other task contexts

The cream-into-tea error illustrates how, by the present account, slips of action stem from representational underspecification. When internal representations of task context are distorted, they can come to occupy a point in representational space intermediate between points the system has assigned to familiar task contexts. The behavior resulting from these ambiguous representations then depends on their proximity or resemblance to the latter reference-like points.

Underspecification and frequency-biasing

In his discussions of underspecification and action slips, Reason (1979, 1992) emphasized that, when task representations are underspecified, behavior tends to default to high-frequency responses. For example, when degraded task representations lead to a lapse from one task into another, this tends to involve a shift from a less frequently performed task into a more frequent one.

In our simulations, we asked whether this frequency effect would also appear in the behavior of the model. This involved varying the relative frequencies with which the model was exposed to the coffee and tea tasks during training, and evaluating whether this affected the frequency of the cream-into-tea error. The model did in fact reproduce the empirical phenomenon; lapses from tea-making into coffee-making (the cream-into-tea error) were more frequent when, during training, the coffee task had been presented relatively frequently (detailed results are presented by Botvinick & Plaut, 2002).

The factors underlying the frequency effect, as it emerges in the model, derive directly from the model's use of a shared, continuous space for task representation. Consider once again the schematic in Fig. 4. Note that, in order for the cream-into-tea error to occur, it is not necessary for the distorted context representation to precisely overlap what we have been calling the coffee pattern. Instead, distortion need merely bring the model's context representation into the coffee pattern's vicinity. This is because, as the model assigns specific points to particular task contexts, the regions surrounding these locations develop similar associations. Thus, in the space the model uses to represent task context, the coffee and tea contexts correspond not only to specific points, but also to entire regions. In order to cause the cream-into-tea error, distortion need only displace the model's context representation across the boundary between the two relevant regions. The idea is illustrated schematically in Fig. 4, using a dotted line to represent the boundary.

The effect of task frequency emerges because the relative frequency of different tasks during training influences the way that learning "carves up" the space used for context representations. Larger portions of the space are apportioned for more frequent tasks. This, in turn, influences where the boundaries between regions fall. Consider Fig. 4 once again. If tea-making were presented more often during training than coffee-making, this would have the effect of expanding the region dedicated to the tea task and shrinking the region dedicated to coffee-making. In the schematic, this would be reflected in a shift of the dotted line to the right. Training more frequently on coffee-making would have the opposite effect, leading to a leftward shift of the boundary. This shifting of boundaries is the link between task frequency and error rates. As coffee-making becomes more frequent and the boundary between the coffee and tea regions shifts leftward, it becomes easier for noise to displace the model's context representation "over the line" into the coffee region, causing the cream-into-tea error. To state the point more generally: The less frequently a task is performed, the smaller a region it occupies in the space used for task representations, with the result that distortions can more easily displace the model's context representations out of this region into one used to represent a different context, resulting in an error.

Actual data from the model, illustrating the effect of frequency on task representations, are shown in Fig. 6. The plots illustrate the effect of gradually distorting the model's internal representations, moving through representational space from what we have been calling the tea context to the coffee context (one can picture this as the traversal of an imaginary line running between the two solid points in Fig. 4). The traces in each plot show the effect of this gradual distortion on the activation values of two output units: (1) the unit representing the action that would be correct in the tea context (*put-down*), and (2) the unit representing the action that would be correct in the coffee context (*locate-cream*, also the first step in the cream-into-tea error). As the model's context representation is gradually distorted, the activation of *put-down* gradually falls and that of *locate-cream* rises. The point at which the two traces cross can be thought of as corresponding to the dotted line in Fig. 4, the boundary across which noise must transport the model's context representation in order to produce an error. As Fig. 4 shows, task frequency affects the location of this cross-over point. When tea-making is more frequent during training (Fig. 6, top), the point falls toward the right, indicating that a greater share of representational space has been apportioned to the tea task. When tea-making is relatively infrequent (bottom), the cross-over point shifts to the left, making it easier for representational distortion to cause a lapse from tea-making into coffee-making.

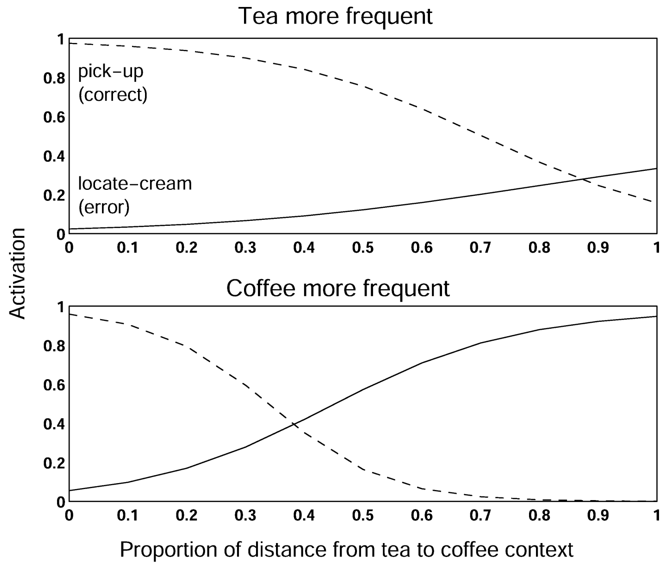

Fig. 6 Simulation data, showing the effect of gradually distorting the model's context representation away from the tea pattern and toward the coffee pattern, on the step just following the completion of sugar-adding (i.e. the step where the cream-into-tea error, if committed, begins). The *x*-axis indicates the proportion of the distance between the two reference representations has been traversed. The *dashed trace* shows the activation of the *put-down* action unit (the correct action on this step), the *solid trace* the activation of *locate-carton* (the first action in the cream-into-tea error.) *Top*: result of presenting tea-making five times as often coffee-making during training. *Bottom*: result of presenting coffee-making five times as often as tea-making. For details of simulation and analysis, see Botvinick & Plaut (2000)

Information-sharing

So far, we have considered the implications of graded, distributed task representation for the notion of representational underspecification. We now turn to a second pivotal idea, drawn from existing theories of action. This is the idea that there is *information-sharing* among the representations underlying interrelated tasks. The concept has been discussed at length by Schank & Abelson (1977; see also Schank, 1982). They suggest that closely related activities rely on common underlying mental representations. One example they provide concerns the activity of eating in a restaurant. As they point out, an individual may be acquainted with a variety of different types of restaurant: fancy restaurants, coffee shops, fast-food restaurants, cafeterias, etc. By their account, however, behavior in all of these settings would be guided by a single, general representation of the restaurant context (for related accounts, see Grafman, 1995; Minsky, 1975; Rumelhart, 1980). This general representation can, by the Schank & Abelson (1977) account, be fine-tuned to specific restaurant settings, allowing the representation's basic elements to be "re-used" as the system implements closely interrelated procedures.

The idea of information-sharing is difficult to square with models that portray task representations as discrete, non-overlapping entities. It is true that, within such models, multiple task units may send connections to a given subtask unit, allowing for a limited, piecewise variety of information-sharing (an issue to be discussed further below). However, the notion of information-sharing as introduced by Schank & Abelson (1977) and others since involves a broader claim, which is that individual representations of task context can be parameterized or tuned in order to implement different tasks, or different versions of the same task. Theories that have attempted to implement information-sharing in this sense, including the theory presented by Schank & Abelson (1977), have tended to rely on elaborate, ad hoc constructs, sometimes rivaling in complexity the phenomena they purport to explain. In contrast, if task representations are viewed as graded, distributed patterns, information-sharing arises as a natural consequence. In order to unpack this point, the following sections examine the role of information-sharing in our model of coffee-making.

Information-sharing using graded context representations

A preliminary indication of the role of information-sharing in the model was provided in Fig. 3. As discussed earlier, this diagram represents the context representations arising during a single subtask (sugar-adding) as performed in different contexts (coffee- versus tea-making). The differences between the two trajectories indicate that the patterns are "shaded" in order to indicate which task is being performed. Despite these differences, though, the two trajectories remain very similar to one another. It is this similarity that indicates the presence of information-sharing; it indicates that the system has "re-used" patterns to represent interrelated operations. Note that the system's use of a continuous representational space allows it to re-use patterns without precisely duplicating them. Because its task representations can be similar without being identical, they can respond to the overlap between tasks while also responding to the differences between them.

Of course, in their discussion of information-sharing, Schank & Abelson (1977) were concerned not with the performance of a single subtask in different contexts, but with the performance of different (though related) task sequences. To illustrate how the model behaves in such a case, we reimplemented the model reported by Botvinick & Plaut (2002) but added a new task, cocoa-making. In our implementation, as in the real world, making cocoa shares a great deal with another action sequence, adding sugar to coffee from a sugar bowl. Both tasks involve the same basic sequence of actions: A spoon is used to scoop up powder from a container, pour that powder into a liquid-filled cup, and then stir the liquid. Of course, at other levels, the two tasks are not identical. They differ in terms of the appearance of the objects involved, for example the difference in color between sugar and cocoa-mix. Training the model simultaneously on these tasks thus faced the model with two distinct but interrelated procedures, providing an opportunity for information-sharing of the kind described by Schank & Abelson (1977).

Figure 7 shows the context representations the trained model used in performing cocoa- and sugar-adding, as visualized with MDS. Once again, as in Fig. 3, there is an obvious resemblance between the two trajectories. Here again, the model uses similar patterns to represent similar tasks, implementing a form of information-sharing. However, again as in the previous example, the re-use of patterns is only approximate, allowing the differences between the cocoa and sugar procedures to be acknowledged as well as their similarities. The model's use of a similarity-based representational scheme allows it to implement a form of information-sharing that is responsive to both the similarities and differences between tasks.

Making inferences about action

Schank (1982) argued that one role of task representations might be to support inferences about action in partially novel situations. For example, he suggested that the representations developed through one's experiences eating at McDonald's would provide a basis for knowing how to act when visiting a Burger King for the first time (see Schank, 1982, p. 24). It is worth noting that the mechanisms implemented in the model we have been discussing support this kind of generalization. For

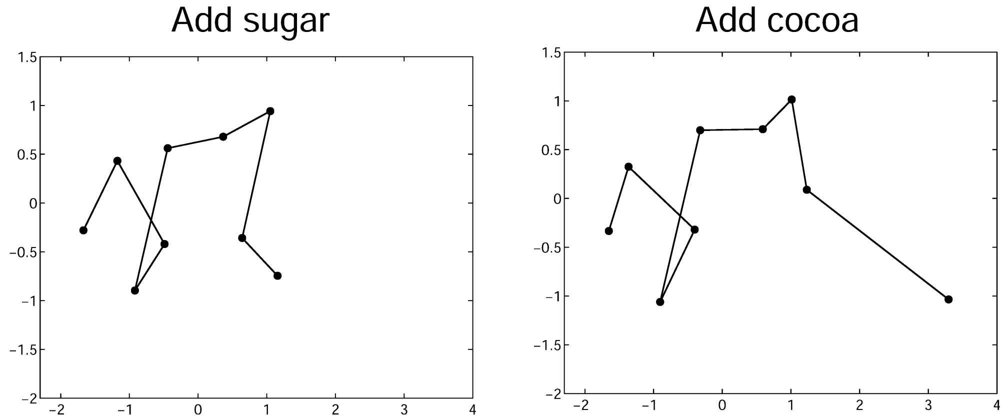

Fig. 7 Internal representations arising during performance of sugar-adding (*left*) and cocoa-adding (*right*) subroutines, visualized using multidimensional scaling

example, when trained on coffee- and tea-making (but not cocoa-making), and then presented with inputs representing a container filled with cocoa mix, the network inferred the proper response sequence, locating and picking up the spoon, scooping powder from the container, and pouring it into the cup. This behavior can be attributed only to the model's previous training on sugar-adding. In fact, when MDS is used to visualize the internal representations arising during the inferred cocoa-adding sequence, they resemble the representations associated with sugar-adding (some relevant quantitative data will be presented below).

In this example, the model's inferences about action turn out to be correct; cocoa-adding can be performed using the same actions as sugar-adding. However, as discussed in the next section, the situation becomes considerably more interesting when the network's inferences turn out to be incorrect, and the network is expected to learn from its mistakes.

Learning

Some of the richest implications of graded, distributed task representations emerge in the context of learning. This is especially true when the task to be learned shares structure with a task that is already familiar. In this section we examine the processes at work in the model when faced with this situation.

Learning through violated expectations

In his work on task representation, Schank (1982) proposed that knowledge structures for action develop in direct response to the violation of expectations. The suggestion is closely tied to Schank's (1982) points concerning inference-making. In short, in novel situations, the processing system makes guesses about appropriate actions by applying knowledge relating to similar contexts, and learning occurs when these predictions are violated.

Like other key assertions made by Schank (1982), this one turns out to describe well the functioning of the connectionist model we have been discussing. In order to illustrate, we again presented a model that was already trained on coffee- and tea-making with the new task of making cocoa. This time, however, there was a new twist. Before, the target actions in cocoa-adding had precisely matched those involved in using the sugar bowl, allowing the network to infer the correct actions for cocoa-making even without specific training. In this follow-up simulation, the target actions in the cocoa task were slightly changed. In the previous version, the action of spooning out some cocoa mix had been represented with a single target output unit (the same *scoop* unit included in the sugar-bowl sequence). Now, on the same step, the target action involved activating two output units, the same *scoop* unit, but now also a "modifier" unit labeled *big*. Together, these units were meant to indicate that spooning out cocoa mix should involve taking a bigger scoop than the one used when spooning out sugar for coffee or tea.

When faced with this new version of cocoa-making, the model's responses were at first incorrect. As in the previous simulation, the network "guessed" that it should use precisely the same actions in adding cocoa that it had learned for adding sugar, failing to activate the *big* unit along with the *scoop* unit. However, with continued training, the network soon began producing the correct "big scoop" when executing the cocoa task. The learning process, as in Schank's (1982) theory, is driven by the violation of predictions. In the model learning occurs through the gradual adjustment of connection weights. These, in turn, are driven by an error computation; actual outputs are compared with desired outputs, and weight adjustments are made in order to reduce the difference between the two (see Rumelhart, Durbin, Golden, & Chauvin, 1996). Thus, in the present example, the learning process that ultimately allows the network to perform the cocoa task correctly is driven almost entirely by events occurring on one step of processing, the step where the model predicts the *scoop* action and this "expectation" is contradicted.

With these points about learning as context, we now return to the issue of task representation. The foregoing discussion raises the question: How do the model's task representations change through learning?

More specifically, how does the system arrive at appropriate representations for new tasks? As discussed in the next section, it is here that the implications of a graded, distributed representational scheme become most apparent.

Learning as a remodeling of task representations

It is informative to consider how the cocoa task we have just introduced might be implemented in a model using discrete, non-overlapping task representations. One way of accommodating the new task would be to assign to it a previously uncommitted, and unconnected, task unit (for a relevant account, see Houghton, 1990). However, this would not acknowledge the close resemblance between the cocoa-adding task and the previously acquired sugar-adding task. Adding a new task unit fails to capitalize on existing knowledge in the way that Schank (1982), for example, described. The only evident alternative, given the assumptions of traditional models, is to use the already existing sugar-adding representation to also cover the cocoa task. But, given this approach, how is the difference between sugar-adding and cocoa-adding (the size of the scoop) to be represented?

The dilemma described here arises as a direct consequence of the assumption that task representations are discrete and independent. In any system relying on such representations a categorical decision must be made, in coping with any pair of tasks, as to whether those tasks will be subserved by a single task representation or two independent task representations. A corollary of this is that, during learning, the system must decide whether or not a new task merits the creation of a new task representation, and if so at what stage of learning this act of creation should occur.

A system using graded, distributed task representations does not face these decisions, or the conflicts they can create. Since there is information-sharing between task representations, the system can capture the distinctions between tasks while still capitalizing on structural overlap. Furthermore, during learning, the system need not create new task representations "from scratch." Since task representations occupy a continuous representational space, new representations can differentiate gradually from established ones.

This latter point is worth illustrating through an example, and for this we return to the second cocoa-making simulation. Here, the model faces a situation where it must learn to represent a task that is similar but not identical to a task it has already acquired. How does the model arrive at a set of representations for the new task? Rather than creating totally new representations in order to handle the cocoa task, the network gradually "remodels" the representations it has already established for sugar-adding. An indication of this process is provided by Fig. 8. The x-axis of this diagram indexes the steps in the sugar and cocoa sequences. The y-axis indicates distance within representational space. Each trace in Fig. 8 shows the distance

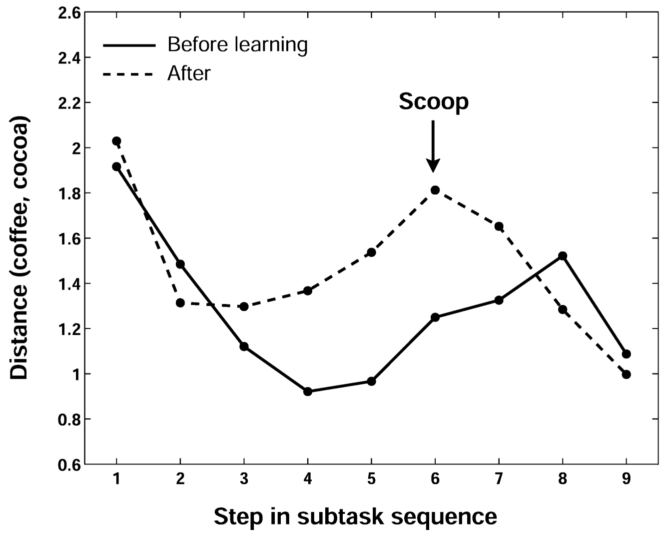

Fig. 8 Reshaping of context representations through learning. The x-axis indexes the steps in the sugar- and cocoa-adding sequences. Each trace shows the Cartesian distance, on each step, between the internal representations arising during sugar- and cocoa-adding. *Solid trace*: before training on cocoa task (at this stage, the model fails to activate the *big* unit along with the *scoop* unit on step six. *Dashed trace*: after training

between two series of internal representations: (1) the series associated with sugar-adding and (2) the series associated with cocoa-adding. The solid line shows the situation before the network has been trained on cocoa-adding. The cocoa-making representations here are the ones arising as the network "infers" which actions to perform, failing to activate the *big* unit along with the *scoop* unit on step six. (The non-zero distance values reflect the fact that the network is using slightly different representations for the cocoa task than the sugar task, even though producing precisely the same actions.) The dashed line indicates the situation after the network has learned to perform cocoa-making using a big scoop. There is increased distance between the sugar and cocoa traces, indicating that the sets of representations used by the model for the two tasks have migrated away from one another in representational space. This distancing is most pronounced on step six of the sequence, the scoop step on which the two sequences differ.

Instead of needing to create a completely new set of representations for the new task, the model can gradually shape new representations, using established representations as raw material. Because this shaping process takes place within a continuous representational space, there is no need for the system to choose when it will "create" a new task representation. New representations emerge bit by bit, through a gradual process of differentiation.

General discussion

Our goal in this article has been to detail an alternative view of the representation of task context. Several currently influential theories portray task representations as discrete, independent, and non-overlapping entities. In contrast, we have suggested that such representations might instead be thought of as graded, distributed patterns occupying a shared, continuous representational space. In other studies (Botvinick & Plaut, 2002), we have shown how this view can be used to account for specific empirical data pertaining to human behavior, both normal and impaired. Here, our goal has instead been to examine the account's fit with some previous proposals concerning the nature of task representations. Specifically, we have focused on two fertile ideas from the literature: representational underspecification and information-sharing. The account of task representation we have presented affords a new perspective on these two ideas, showing in particular how they can both be derived from more basic assumptions about the representation of procedural knowledge.

Graded task representations: a double-edged sword

A key aspect of the present account is that it portrays task representations as responding to the similarities among different task contexts. This point is basic to our account of representational underspecification and action slips, since it makes the system vulnerable to confusions between interrelated contexts. It also provides the basis for our model's account of information-sharing, allowing task representations to acknowledge the overlap between interrelated operations.

In fact, by the present account, underspecification and information-sharing turn out to be two sides of a single coin. The same factors that make the processing system vulnerable to error also give it its representational power and ability to generalize. At this level, the account we have presented answers the intuition, voiced intermittently in earlier writing on action (e.g. James, 1890; Norman, 1981; Reason, 1990), that action errors are somehow tied to the encoding of similarities in task context.

Traditional accounts: some further considerations

Throughout the preceding discussion, we have contrasted the approach taken in our own work with a traditional framework, within which task contexts are represented independently. One basis for this comparison has been to consider the fit of each account with the constructs of information-sharing and representational underspecification. As we have emphasized, both of these constructs, according to their original conception, deal with the intrinsic properties of individual task representations. It is on this level that the ideas of underspecification and information-sharing resonate most strongly with the account we have presented. In contrast, it seems difficult to map the two ideas onto the simple, monadic representations of task context employed in traditional models.

Of course, the fact that information-sharing and underspecification do not arise, in traditional models, at the level of individual task representations does not mean that such models do not engage these issues at all. Indeed, we have already noted ways in which one might argue for the presence of both factors: One can see a version of information-sharing in the fact that multiple task nodes can link to a single subtask node, and one can see underspecification in system states where multiple task nodes are simultaneously active. Earlier, we set these points aside on the grounds that they related to interactions among task representations rather than to intrinsic properties of such representations. However, as we shall now discuss, even if one pans back from the issue of representation, the versions of information-sharing and underspecification implemented in traditional models still suffer from significant limitations.

Consider first the point that, in traditional models, lower-level nodes may receive connections from more than one higher-level node. Certainly, there is a re-use of representations here. However, upon close scrutiny the arrangement appears to carry less representational richness and flexibility than the idea of information-sharing implies. Perhaps the most important problem with the arrangement is that it limits information-sharing to situations where there is, across two contexts, a precise match between clearly bounded segments of behavior. To put it another way, higher-level representations may share information only when the sequences they share are absolutely invariant with respect to context. Thus, coffee- and tea-making may use the same sugar-adding node only if sugar-adding is executed in precisely the same way in the two tasks. Hierarchical, node-based models face difficulty when an operation must be performed differently depending on the larger behavioral context, as, for example, if one preferred to add more sugar to coffee than to tea.

Note, secondly, that the tractability of the node-sharing approach depends on the assumption that tasks are structured at a few discrete levels, usually thought of as the levels of task, subtask and action. A close look at the pattern of shared structure among naturalistic behaviors suggests that the situation is often more complex than this. For example, many activities involve the following sequence: (1) look toward an object, (2) reach for it, (3) grasp it, (4) look to another location, (5) move the hand to that location, and (5) put the object down. Since this subpattern occupies neither the level of individual actions nor that of complete subtasks, it raises an awkward question for a hierarchical, node-based approach: When does such a subpattern qualify for its own schema node? Furthermore, once a commitment to discrete levels of structure is abandoned, how is it determined which schema nodes should compete with one another? Typically, nodes within hierarchical models compete only with other nodes occupying the same level. Any attempt to acknowledge graded, intermediate levels of structure would thus lead to implementational difficulties.

Related issues pertain to the implementation of information-sharing found in production-system models. Many such models (e.g. Anderson & Lebiere, 1998) include an argument-binding or "slot-filling" mechanism that allows productions to be applied to a number of analogous situations. This certainly allows for some degree of information sharing, and a form that (unlike what is found in node-based models) derives from the properties of individual task representations. However, as in the preceding case, difficulties arise when the goal is to apply the approach to structurally rich naturalistic task domains. The basic problem is that, despite the flexibility slots confer, there is still the need to decide, for any two operations, whether they should rely on a single slot-equipped production or instead on two independent productions. For illustration, imagine that a production system is to be used to model the simple task of making a peanut-butter and jelly sandwich. Given that spreading peanut-butter and spreading jelly involve very similar actions, it seems reasonable to posit a single "ingredient-spreading" production, with a slot used to specify the topping. However, difficult questions arise if a wider range of tasks is to be added to the model's repertoire. For example, should the same ingredient-spreading production be applied to the task of spreading sauerkraut on a hot dog? What about spreading icing on a cake, or wax on car? The formalism requires categorical decisions to be made about task-relatedness, when tasks may in fact share graded, multidimensional similarity relations. No such black-and-white decisions are necessary in a system using graded, distributed task representations.

Let us turn now to the issue of representational underspecification. While traditional, node-based representations of task context are not themselves subject to underspecification, one might argue that underspecification can arise in models based on such representations when there is simultaneous activation of multiple task nodes. Indeed, if one changes the unit of analysis from the individual task unit to the entire population of such units, one might well speak of task contexts being represented by the pattern of activity across that population, and this pattern could in turn be understood as a point in a continuous, multi-dimensional representational space. Under this reinterpretation, underspecification in traditional models may appear to involve just the same factors as it does in the model we have presented: a representation of context that lies intermediate between points used to represent familiar tasks. However, the two accounts are not, in fact, identical. The primary difference stems from the fact that, in a node-based model, the point in representational space corresponding to each familiar task lies at precisely the same distance from every other such point. In contrast, in the recurrent connectionist model we have described, representations for interrelated tasks lie closer to one another than they do to those for less similar tasks.

This point has implications for the impact of underspecification on behavior. Specifically, it means that errors based on underspecification will tend to involve a lapse from one task into one that is *closely related* to the first. In order to make the point more directly, let us return to the domain of food preparation, and consider a behavioral repertoire including coffee-making, tea-making, and a less closely related task such as sandwich-making. In the framework we have put forth, the context representations for the two beverage-making tasks would be more similar to one another than either one would be to the representations involved in sandwich-making. As a direct consequence, degradation to context representations during performance of coffee-making would be more likely to result in a lapse into tea-making than into sandwich-making. In a node-based account, in contrast, there would be no more similarity between the task representations for coffee- and tea-making than between either of these and sandwich-making. Thus, if noise were added to the system's task nodes during coffee-making, there is nothing about the way that tasks are represented that would bias the system toward lapsing into tea-making rather than sandwich-making. Of course, patterns of similarity between environmental inputs may go some distance toward creating a bias of this sort. However, such "stimulus capture" will go only so far in explaining available empirical data; existing error corpora provide many examples of slips where a similarity between tasks, rather than a similarity between environmental triggers, appears critical. To take just one example, consider the case of the office worker reported by Reason (1990), who answered the telephone by shouting "Come in!" Admittedly, it is impossible to rule out a role for stimulus capture in this case (perhaps the individual was looking at the door when answering the telephone). However, it seems likely that the similarity between the task of answering the door and that of answering the telephone, a similarity presumably reflected in the mental representations subserving each, also contributed to the slip.

The present discussion has been aimed at highlighting some limitations on the ability of a traditional, node-based account to fully engage the issues of information-sharing and underspecification. However, it is important to emphasize that these are not the only problems that arise in the effort to build models of action based on independent, non-overlapping task representations. Several other more basic issues were discussed in the introduction, and have received a more detailed treatment elsewhere (Botvinick & Plaut 2002).

Limitations of the present account

One important difference between the account of task representation we have put forth and most previous accounts is the central role it accords to learning. Traditional models rely on task representations that are stipulated by the modeler, based on a priori assumptions about task structure. In contrast, representations of context in the present account emerge based on experience, leading to sequencing mechanisms that are tuned to the fine structure of the task domain and that are sensitive to patterns of similarity among different activities. However, while the model's account of learning is one of its strengths, it also carries liabilities. Gradient-descent learning algorithms of the kind employed in our simulations, despite their power, depend on interleaved training in order to prevent so-called catastrophic interference (see McCloskey & Cohen, 1989). While proposals have been made concerning how this problem may be overcome within the nervous system (McClelland, McNaughton, & O'Reilly, 1995), the issue continues to be debated in the literature (see, e.g. Page, 2000).

While it seems reasonable to assume that much learning about sequential tasks is acquired gradually and implicitly, it is also undeniable that humans are able to assemble representations for novel tasks quite rapidly, for example based on verbal instructions. This may seem to provide a challenge for the gradualist account of learning implemented in our model. However, in this case, the problem may be more apparent than actual. As St John (1992) has shown, the same kind of processing mechanisms involved in our model can also be used to rapidly encode and parse linguistic input (see also McClelland, St. John, & Taraban, 1989). In this work, the task posed to the model was to respond to queries about presented sentences (e.g. identifying role-fillers). However, the model could, in principle, have been asked to produce sequential outputs like those produced by the model in our simulations.

The work we have presented deals with knowledge that is essentially procedural; the task representations we have discussed serve only to guide performance on-line. However, there appear to be other ways in which knowledge about tasks can be accessed and applied. For example, planning in unfamiliar or problematic domains may call for a manipulation of task representations quite unlike that involved in stepping through a familiar task. Interestingly, the most successful models in this domain to date have employed independent, non-overlapping task (or goal) representations (e.g. Anderson & Lebiere, 1998; Laird, Newell, & Rosenbloom, 1987). Whether this approach is necessary or optimal presents an interesting question for future research.

A note on the behavioral domain

Discussions of task set, including many of the articles in this special issue, focus on simple laboratory tasks, typically unfamiliar to the experimental participant. We have focused here on a different behavioral domain, that of everyday routine sequential behavior. Needless to say, both areas of behavior are worth understanding on their own terms. However, it is worth noting that certain key issues pertaining to task representation are brought out more strongly when considering naturalistic action. Central among these is the role of similarity. Routine sequential behavior throws this issue into relief because it provides a domain rich in overlap and shared structure.

There is an analogy here to the domain of semantic knowledge. Experiments concerned with this topic tend to employ stimuli drawn from everyday life, tapping into the graded and multidimensional similarity relations that structure our knowledge of everyday objects. It is interesting to note that there is, in fact, an historical and conceptual link between contemporary theories of semantic memory and contemporary theories of action, evident in the hierarchical tree structures frequently invoked in both domains. Given this link, it is perhaps not surprising that recent work in semantics has suggested a reformulation very closely related to what we have suggested here for task representation (see McClelland & Rogers, 1997).

Of course, as we have just noted, it is an open question to what extent the points we have made in the context of routine sequential action might apply to non-routine tasks, and this comment extends to the kinds of task typically used in laboratory studies of task set. Even in the domain of routine action, further empirical work will be necessary if a strong case is to be made for the account we have presented here. Still, it already seems clear that, in seeking to understand task set, it may be helpful to consider a wider range of theoretical possibilities than those currently prevalent in work on action.

Acknowledgements 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, 1999), http://www.cs.cmu.edu/~dr/Lens/.

References

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

Botvinick, M., & Plaut, D. C. (2002). Doing without schema hierarchies: a recurrent connectionist approach to routine sequential action and its pathologies. (submitted).

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

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

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

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.

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* (pp. 337–368). New York: New York Academy of Sciences.

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* (pp. 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*, 1–25.

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

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

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.

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

McClelland, J. L., & Rogers, T. T. (1997). A PDP account of basic category effects and semantic dementia. In *38th Meeting of the Psychonomics Society Abstracts*.

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

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why are there complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review, 102*, 419–457.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: the sequential learning problem. In G. H. Bower (Ed.), *The psychology of learning and motivation* (vol. 24, pp. 109–165). New York: Academic Press.

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

Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision* (pp. 211–277). New York: McGraw-Hill.

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

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

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

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. T. (1990). *Human error*. Cambridge: Cambridge University 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* (pp. 71–91). New York: Plenum Press.

Rohde, D. (1999). *Lens: the light, efficient network simulator*. Technical Report CMU-CS-99-164, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA (available at http://www.cs.cmu.edu/~dr/Lens/).

Rumelhart, D. E. (1980). Schemata: the building blocks of cognition. In Spiro, R. J., Bruce, B. C., & Brewer, W. F. (Eds.), *Theoretical issues in reading comprehension* (pp. 33–58). Hillsdale, N.J.: Erlbaum.

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., 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* (pp. 533–566). Hillsdale, N.J.: Lawrence Erlbaum.

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, N.J.: Lawrence Erlbaum Associates.

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*, 13–28.

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.

St John, M. F. (1992). The story gestalt: a model of knowledge-intensive processes in text comprehension. *Cognitive Science, 16*, 271–306.

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