THE QUARTERLY JOURNAL OF EXPERIMENTAL PSYCHOLOGY, 1993, *46A* (1) 83–118

Finite-state Automata: Dynamic Task Environments in Problem-solving Research

Axel Buchner and Joachim Funke

Bonn University, F.R.G.

This paper presents a new research paradigm for analysing human learning in dynamic task environments based on the theory of finite-state automata. Some of the advantages of the approach are outlined. (1) It is possible to design classes of formally well-described dynamic task environments instead of idiosyncratic microworlds that are difficult if not impossible to compare. (2) The approach suggests assumptions about the mental representation of a discrete dynamic system. (3) The finite-state automata formalism suggests systematic and appropriate diagnostic procedures. (4) Using finite-state automata to design dynamic task environments, one does not have to give up the "ecological validity" appeal of computer-simulated scenarios.

An experiment on the utility of an external memory support system with system complexity and type of memory support as independent variables is reported to illustrate the application of the formal framework. Systematically derived dependent variables reflect both system knowledge and control performance. The results suggest that the benefits due to the availability of the external aid vary as a function of the complexity of the task. Also, using reaction time measurements, priming phenomena have been found that point to the importance of sequentiality in the representation of discrete systems. It is concluded that the approach, although not entirely new in experimental psychology, awaits further exploration in research on human learning in dynamic task environments and promises to be a stimulating paradigm for both basic and applied research.

Requests for reprints should be sent to Axel Buchner or Joachim Funke, Department of Psychology, Bonn University, Römerstraße 164, D-5300 Bonn 1, F.R.G. E-Mail: UPS11B@ibm.rhrz.uni-bonn.de [Buchner], UPS141@ibm.rhrz.uni-bonn.de [Funke].

This research was supported by a grant from the Commission of the European Community (ESPRIT Basic Research Action #3219). We would like to thank Donald E. Broadbent (Oxford), Edgar Erdfelder (Bonn), Jens Beckmann (Leipzig), and two anonymous reviewers for their very valuable and helpful comments on an earlier draft of this paper. We are also grateful to Lothar Schmitt for programming MacFAUST, and to Beate Nikelowski for her assistance with data collection.

The purpose of this paper is twofold: (1) We wish to introduce human interaction with finite-state automata as a new research paradigm for studying processes of knowledge acquisition and knowledge application in dynamic task environments. (2) In order to illustrate the approach, a study will be reported in which finite-state automata were used as dynamic task environments. The study aimed at investigating the usefulness of an external memory aid in exploring and controlling a device.

Using computer-simulated scenarios in problem-solving research has become increasingly popular during the last decade (e.g. Berry & Broadbent, 1984, 1987, 1988; Brehmer, 1987; Broadbent, Fitzgerald, & Broadbent, 1989; Dörner, 1987; Funke, 1988, 1991; Hayes & Broadbent, 1988; Hoc, 1989; Hunt & Rouse, 1981; Jeffries, Polson, & Rasran, 1977; Moray, Lootsteen, & Pajak, 1986; Morris & Rouse, 1985; Plötzner, Spada, Stumpf, & Opwis, 1990; Putz-Osterloh & Lemme, 1987; Sanderson, 1989). This approach to problem solving seems attractive for several reasons. In contrast to static problems, computer-simulated scenarios provide the unique opportunity to study human problem-solving behaviour when the task environment changes concurrently. Subjects can manipulate a specific scenario via a number of input variables (their number typically ranging from 2 to 20, and in some exceptional instances up to 2000), and they observe the system's state changes in a number of output variables. In exploring and/or controlling a system, subjects have to acquire continuously and use knowledge about the internal structure of the system.

Computer-simulated "microworlds" seem to possess what is called "ecological validity". Simulations of (simplified) industrial production (e.g. Moray et al., 1986), medical systems (e.g. Broadbent, Berry, & Gardner, 1990), or political processes (e.g. Dörner, 1987) have the appeal of bringing "real-world tasks" to the laboratory. This has stimulated the use of a great diversity of dynamic systems as experimental task environments, each of which is designed to relate to a different aspect of "reality". The problem, however, is that such vastly different experimental tasks, and, hence, the results of experiments using these tasks, are very difficult to compare. In particular, it becomes unclear as to whether one should attribute experimental findings to the experimenter's manipulation or, rather, to the peculiarities of the task employed. Most systems differ not only with respect to surface features (i.e. the semantics implied by the labelling of their input and output variables), which we know to have strong influences on problem-solving behaviour in both static tasks (e.g. Kotovsky & Fallside, 1989; Novick, 1988; Wagenaar, Keren, & Lichtenstein, 1988) and dynamic ones (e.g. Hesse, 1982; Putz-Osterloh, 1990). Equally important, for most systems it is unclear how one should compare them with respect to the underlying formal structure.

There are two possible solutions to the latter problem. One possibility is to define a set of formal dynamic system characteristics and use this set

for systematically comparing the tasks used in various experiments (e.g. Funke, 1990). Such an analysis will at least give a rough idea of whether or not two dynamic tasks could yield comparable results. The other possibility is to derive different dynamic task environments from the same formal background. The formal homogeneity of different task environments facilitates comparisons between experiments and increases the chances of discovering effects that are not only "local".

The theoretical framework we refer to is the cybernetic theory of finite-state automata (cf. Ashby, 1956; Hopcroft & Ullmann, 1979; Roberts, 1976; Salomaa, 1985). We wish to show that, from the perspective of cognitive psychology, the paradigm of investigating human interaction with finite-state automata has several interesting aspects. These are mentioned here and discussed in greater detail further on. (1) The theory of finite-state automata may serve as a basis for constructing *classes of formally well-described dynamic task environments*. As a consequence, it becomes possible to construct different problems that may share well-known properties and differ with respect to a critical feature. (2) The formal description of finite-state automata suggests interesting assumptions about plausible *cognitive processes and forms of mental representation* necessary to control a discrete dynamic system effectively. (3) The same formalism suggests appropriate and systematic *diagnostic procedures* that closely correspond to the assumptions about mental representation. (4) Using finite-state automata, one does not have to give up the "*ecological validity*" appeal of more conventional dynamic task environments.

Formal Background

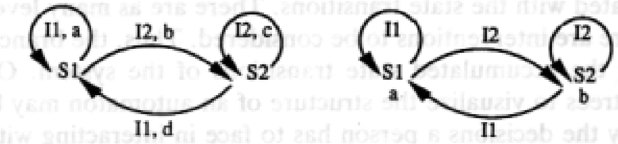
Finite-state automata theory is a well-elaborated framework in the area of computer science. Here, however, we make use only of the framework's most elementary concepts. Within finite-state automata theory, any system can be defined and exhaustively described by a transformation function that specifies the state transitions given a specific state of the system. In this paper we focus on deterministic automata. A deterministic finite-state automaton is defined by three sets and two functions:

1. a finite set X of input signals (the input alphabet);
2. a finite set Y of output signals (the output alphabet);
3. a finite set S of states;
4. a transition function δ, which is a mapping of $S \times X$ on S and which determines the next state of the system as a consequence of the input signal;
5. a result function λ, which is a mapping of $S \times X$ on Y and which determines the output signal of the system as a consequence of the input signal.

The automaton $A = [X, Y, S, \delta, \lambda]$ is called a *deterministic Mealy-automaton* (see Figure 1). To make things more concrete, the input alphabet of an arbitrary technical device consists of the button and switch positions that can be selected as input at a certain point in time. The output alphabet contains all possible display settings. It is assumed in the above definition that the system works on the basis of a discrete time scale. At each point in time, the system is in a certain state in which it receives exactly one *input signal* (e.g. on a VCR, the "fast forward" button is pressed). The system then moves to the *next state*, which is determined by the transition function δ (e.g. the VCR starts to wind the video tape). Subsequently, the device emits exactly one *output signal*, which is determined by the result function λ as a consequence of the current state and the input signal (e.g. the "fast forward" arrows on the VCR's front display are highlighted). Note that in this general version the output signal is informative about both the next system state and the input signal (and, hence, about the previous system state). To illustrate this with another example, an error message of a computer program typically contains information about both the present state and the previous state. Thus, the same state $s \in S$ (e.g. a fatal system error) may be associated with a number of different output signals $y \in Y$ (e.g. error messages), depending on what preceded the transition to that state.

An automaton in which the output signal $y \in Y$ depends only on the new state $s \in S$ as determined by $\delta(s, x)$ (i.e. y is not a direct function of the input signal $x \in X$) is called a *deterministic Moore automaton*. In this case a marker function μ exists, which is a mapping of S on Y, replacing the result function λ. In other words, the output signal contains information only about the state of the system following the intervention (e.g. that a system error has occurred), and not how one got there (e.g. which type of error caused the computer to crash). Numbers displayed on a pocket calculator may also serve as everyday examples for this situation. The digits in the calculator's display do not unambiguously inform us about the calculator's previous state and the last input. Thus, in a Moore automaton, an output signal is less informative, because it only reflects the current state of the system. This is an important system characteristic to keep in mind in constructing dynamic task environments. Naturally, in any realistic automaton both forms of output signals may coexist. Figure 1 illustrates in the simplest possible case the formal difference between Moore and Mealy types of automata.

Two convenient ways of describing finite-state automata are used: state transition matrices and directed graphs. Each possible description of finite-state automata puts a different emphasis on certain aspects of system behaviour. Knowing about these differences is helpful in constructing dynamic task environments. A state transition matrix contains in its cells



	Input Signal of Intervention I_t			
	I1		I2	
State S_t	Next State	Output Signal	Next State	Output Signal
S1	S1	a	S2	b
S2	S1	d	S2	c

(a)

		Input Signal of Intervention I_t	
		I1	I2
State S_t	Output signal	Next State	Next State
S1	a	S1	S2
S2	b	S1	S2

(b)

FIG. 1. Directed graphs and state transition matrices illustrating (a) a Mealy and (b) a Moore automaton. S1 and S2 represent the two states of the automaton, I1 and I2 represent the input signals, and lower-case letters represent the output signals.

the automaton's state at time $t + 1$ (S_{t+1}, the next state), given a specific state at time t (S_t, the current state) and a specific input signal at time t (I_t, the user intervention). In each column it contains the "function" of an input signal, and the rows reflect possible next states given a certain current state.

Figure 1 also shows that every deterministic automaton may be unambiguously described by a directed graph D with $D = (V, OPE)$, where V is a set of vertices (states, nodes) and OPE is a set of ordered pairs of elements of V called arcs (state transitions, edges). For small automata, directed graphs are a particularly useful tool for visualizing the automata's functional characteristics.[1]

Another form of describing the characteristics of a finite-state automaton (not illustrated in Figure 1) is a *tree*. A tree is an ordered graph consisting of a "source" node from which hierarchical "branches" to successor nodes originate. The source node represents the initial state of the system. Successor nodes are all states that can possibly be reached from the current node by any of the available interventions. The "leaves" associated with the branches represent the output signals of the system

[1]In addition, graph theory provides certain (descriptive) concepts for characterizing finite-state automata, such as different forms of connectedness or arc-vulnerability. Roughly, for instance, arc-vulnerability describes the degree to which there exist alternative sets of state transitions that may be used if one set of transitions is no longer available to get a system from a specific state S_t to another specific state S_{t+n}. This may happen, for instance, due to a system failure or because of forgetting on the side of the user. The smaller the number of alternatives, the more vulnerable the system is relative to the S_t–S_{t+n} transition (for further details, see Roberts, 1976).

associated with the state transitions. There are as many levels of branches as there are interventions to be considered. Thus, the branches of the tree reflect the accumulated state transitions of the system. One reason for using trees to visualize the structure of an automaton may be to illustrate quickly the decisions a person has to face in interacting with the system.

The cases considered so far involve only user-generated state transitions. However, state transitions may also occur autonomously (i.e. not caused by direct user interventions). *Autonomous transitions* occur as a function of discrete time intervals. As an everday example of such time-dependent transitions, consider an automatic ticket vendor that ejects the inserted money if no user input occurs within a certain time interval. In order to represent time-dependent transitions, one can simply add a separate column to the transition matrix analogous to a new input signal. This new column contains for each state as parameters not only the next system state S_{t+1} but also the length of the time interval after which the specified state transition will occur (unless, of course, the user selected a different intervention before the end of the time interval).

Transparency as an important system characteristic depends on the nature and the number of *latent states* implemented in a system. A state is said to be latent if, for instance, a state transition to this state results in an output signal identical to the signal of the preceding transition. A ticket vendor that does not emit information about how much money has been inserted can serve as a simple example. After each coin inserted, the system state changes, but the output signal stays constant. Roughly speaking, the larger the number of different states that share the same output signal, the less transparent the system will appear to the person trying to interact with it.

Attractive Features of the Discrete Systems Paradigm

The paradigm of human interaction with finite-state automata has a number of attractive features for studying processes of knowledge acquisition and knowledge application in dynamic task environments. These features, which have already been mentioned, are described here in greater detail.

1. Constructing Classes of Well-described Dynamic Task Environments. This is an important aspect, as manipulating the properties of task environments seems essential for experimental cognitive psychology. Yet in the area of problem-solving research this aspect has often been neglected. Instead, what we find is a collection of simulated scenarios, most of which have been constructed to "mimic" some aspect of reality more or less adequately. These scenarios—and, hence, the results of experiments

employing them—can hardly be compared, and it is difficult to manipulate isolated properties of unsystematically constructed "realistic" scenarios (see e.g. Brehmer, Leplat, & Rasmussen, 1991). One reason for this deficit is that the appropriate formal criteria are not readily available.

Systematically varying and controlling the properties of task environments helps to detect effects that are unique to a specific task, and it may, at the same time, serve to estimate the impact of these properties on processes of knowledge acquisition and application. Within the theory of finite-state automata the tools are available for exhaustively describing different discrete dynamic systems on the basis of the same formalism. This facilitates both system comparisons and systematic variations of single task properties. A concrete example may be system complexity. System complexity is determined by the number of states a device can be in, and by the number of interventions with different consequences possible for a given system state. The number of different interventions corresponds to the number of potential user decisions given that state. For each state, the number of different interventions may vary between 1 (a "trivial" case in which all input signals have the same consequence) and the number of input signals available (each input signal has a different consequence). McCabe (1976), for instance, has introduced a complexity measure that takes into account these parameters that together define the decisional structure inherent in a system. This complexity measure may be used to characterize the overall complexity of automata. Considering the graphical representation G of a finite-state automaton with n states, e edges, and p connected components, complexity C is defined as

$$C(G) = e - n + 2 * p$$

As p is different from 1 only for hierarchically nested automata (a case that is irrelevant for our present purpose), we can say that for the extremely simple example in Figure 1 we find $e = 4$ edges and $n = 2$ states, resulting in $C = 4 - 2 + 2 * 1 = 4$. This figure may then be used to make ordinal comparisons between different automata.[2] McCabe's measure is applied to the automata used in our experiments.

Of course, experimental research using dynamic task environments must not exhaustively focus on the variation of formal properties of task environments. However, formal properties may serve as a *first basis* for interpreting psychologically interesting effects (such as differences in the amount of knowledge acquired) and as a stimulant for interesting experi-

[2] C takes on a minimal value of 1 for automata in which all interventions that are possible at a given state lead to the same next state. The graph of such automata takes on the form of a chain, leading in a straight line from the initial state via all intermediate states to the terminal state. Hence, the number of states n surpasses the number of edges e by exactly 1, thus $C = (n - 1) - n + 2 = 1$.

ments, particularly if subjects' performance deviates from what would be expected on formal grounds.

2. *Plausible Cognitive Processes and Forms of Mental Representation.* The formal descriptions of automata also provide a basis for selecting plausible psychological hypotheses about their mental representation and about processes of knowledge acquisition. The user's knowledge about a system can be described in terms of those parts of the transition matrix that are represented in memory and available for guiding system interventions. We call this the person's "individual transition matrix" (ITM). The person's ITM may, of course, deviate from the automaton's transition matrix because it is incomplete or because it contains incorrect transitions.

If a person is confronted with a previously unknown automaton and begins to explore this device, learning about the functioning of it must begin at the level of individual state transitions composed of a previous state, an intervention, and a next state. A person's experiences of these transitions while exploring the automaton constitute the "entries" for the ITM. At that level, the signals belonging to different states, interventions, and next states must become associated. Figure 2 illustrates the necessary associations between the basic components of individual state transitions. (1) We assume that a system state becomes associated to a specific intervention (F_1) as a consequence of the feedback provided by the subsequent state of the system. Such an association could be the learning to press a stop switch in an emergency situation. (2) The intervention itself may be associated with a specific subsequent system state (F_2). Pressing the off-switch of a device, for instance, will be strongly related to the subsequent terminal state. (3) States may be directly associated to subsequent states (F_3), particularly if there is only a limited choice of possible interventions, if the choice of an intervention does not matter to the state transition, or if the transition occurs autonomously (i.e. as a function of a discrete time

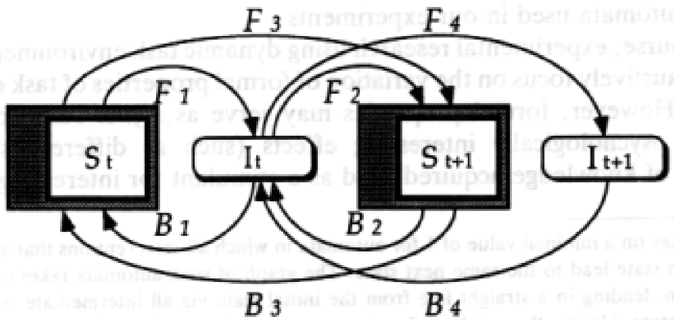


FIG. 2. Forward (*F*) and backward (*B*) associations between basic components of state transitions.

interval without an explicit user intervention). (4) We need to consider associations of subsequent interventions as a component association that should be important when output signals of the system are not attended (F_4). Following manual or cookbook instructions may serve as a prototypical example in this context.

In analogy to paired associate learning findings (cf. Martin, 1965) we may expect both forward and backward associations to be formed (the latter are referred to as B_1 to B_4 in Figure 2). However, forward associations should be dominant, as free exploration of an automaton results in a more serial learning type of experience. In contrast, paired associate learning experiments that find strong backward associations typically randomize the order of presentation of the pairs of stimuli from trial to trial, thus preventing serial learning from playing an important role (e.g. Harcum, 1953; Murdock, 1956, 1958; Richardson, 1960).

Of course, combinations of component associations will be relevant depending on the situation. For instance, associations F_2 and F_3 are relevant for predicting the next system state given the present state and an intervention. In making such a prediction, the current state and a specified intervention may be combined in short-term memory to form a "compound cue" (Gillund & Shiffrin, 1984) to retrieve the next state of the system.

Later on in the process of learning, people may no longer primarily use knowledge about individual state transitions to control a device. Instead, they will cluster individual state transitions into more abstract concepts according to, for instance, the statistical properties of the elements of the state transitions (e.g. co-occurrence of a subset of output signals with a specific subset of input signals) in order to reduce memory load.

We can distinguish two different ways of organizing clusters of state transitions. (1) "Routines" may be developed to get a system reliably from one state to a distant state. This can be referred to as the formation of "*horizontal chunks*" of state transitions. For example, the state transition sequence S_t–I_t–S_{t+1}–I_{t+1}–S_{t+2}–I_{t+2}–S_{t+3} may be reduced to the form S_t–[I_t–I_{t+1}–I_{t+2}]–S_{t+3} where the interventions necessary to get from state S_t to S_{t+3} form one single component of a compound state transition and the user no longer needs to attend to the intermediate output signals (e.g. Anderson, 1982; Frensch, 1991; MacKay, 1982; Newell & Rosenbloom, 1981). (2) State transitions can be combined across a specific intervention or a specific state, given the intervention or the state can be identified as the source of a specific form of invariance. This process can be referred to as the formation of "*vertical chunks*" of state transitions. An example could be an intervention to change the mode of operation of a device (in the most simple case this is an "on/off" switch).

The concepts induced from the experience of individual state transitions are necessarily more abstract in that they no longer correspond to one single specific physical event in the automaton. For instance, concepts like copying, cutting, deleting, and inserting text in a word processor may be grouped as editing functions. In a new dynamic task environment, however, learning-by-exploration will start at the level of individual state transitions, and it seems necessary to understand the process of learning at the level of state transitions before proceeding to a higher level of knowledge organization.

3. Diagnostic Procedures. A frequent practice in problem-solving research is to use specifically designed questionnaires or performance measures that are directly derived from the task at hand, such as the "production output" in an economic scenario. The problem with these idiosyncratic measures is twofold. (1) Again, it is difficult if not impossible to compare such measures if they stem from different tasks and use system parameters as performance criteria. (2) They have no clear relation to "classical" measures of memory, and hence we renounce a considerable body of information accumulated about these latter measures (e.g. Posner, 1978).

The formalism taken from the theory of finite-state automata provides tools for developing adequate and systematic diagnostic procedures. We can assume that a person's experiences of state transitions while exploring an automaton constitute the "entries" for the ITM. State transitions, in turn, consist of a given system state at time t (S_t), an intervention at time t (I_t), and a next system state at time $t+1$ (S_{t+1}). A straightforward way to assess users' representations about a discrete dynamic system, then, is to confront them with two elements of this triple and ask for the missing element. This results in three basic types of questions that can be asked to investigate a given state of knowledge:

1. *Prognostic question*: Given state S_t and intervention I_t, what new state S_{t+1} will result?
2. *Interpolation question*: Given state S_t and state S_{t+1}, what intervention I_t does produce this state transition?
3. *Retrognostic question*: Given an intervention I_t and a resulting state S_{t+1}, what was the previous state S_t?

With these questions it is possible to take "samples" from the ITM. For deterministic discrete systems the answer to Question 1 always has only one correct solution. For Questions 2 and 3, however, the actual answer may be taken from a set of correct items, depending on the specific characteristics of the automaton. The questions may be presented in analogy to classical direct measures of memory, either in a cued recall situation (the person must recall the missing element) or in a recognition procedure

(the person must select the missing element from a list of alternatives). Also, we have been successful at using variants of these questions that constitute indirect measures of system knowledge in that they do not require an explicit recollection of the prior learning episode (Buchner, 1993).

In addition to presenting only two out of three elements, one can also expose subjects to entire state transitions that are either possible or impossible for a given device and ask for a response indicating the correctness of the transition. This is similar to a classical verification task, and both reaction time and error rates provide well-known dependent measures.

A unique feature of dynamic systems tasks is that "control performance" provides an additional access to a person's knowledge about a system. For evaluating these performance data and for making performance comparisons between subjects it is essential to have a criterion for optimal performance. This criterion is directly available within the finite-state automata approach. Given a present state of a discrete system and an arbitrarily defined goal state, it is always possible to specify whether there exists a sequence of interventions to reach the goal state and, if so, how many and what steps constitute an optimal sequence of interventions (i.e. a sequence involving a minimal number of steps).

The finite state automata formalism also suggests other performance measures. For instance, subjects' exploration behaviour (i.e. the way they approach the knowledge acquisition task) may itself be an interesting basis for additional dependent variables. A readily available indicator of exploration behaviour is the number of different state transitions explored relative to all states in the state transition matrix of the system. One can assume that under difficult learning conditions subjects restrict their exploratory activities to a smaller number of transitions to build up firm knowledge about the device.

4. Ecological Validity. Finally, we want to point out that many technical systems we deal with in everyday life are adequately described within the formalism provided by finite-state automata theory. Examples include computer programs, TV sets, programmable VCRs, digital wrist watches, banking machines, and so on (see also the examples given by Weir, 1991). In addition, consider some highly formalized way of social interaction. For instance, everyday experience with administrative processes is that bureaucratic institutions accept only a finite set of input signals, take on a finite set of states, and emit a finite set of output signals. Thus, in drawing upon a well-developed formalism for constructing dynamic task environments, one does not automatically lose the appeal of "ecological validity" that is often demanded of psychological research.

AN ILLUSTRATIVE EXPERIMENT ON THE EFFECTS OF EXTERNAL MEMORY AIDS

To illustrate the approach outlined above, we briefly present an experiment that used finite state automata as dynamic task environments to investigate the utility of an external memory for learning about a system.

Three groups of subjects performed successively on two different unknown automata. For each automaton, their task was the following: subjects were instructed to explore each automaton on two subsequent "exploration phases" and to find out how it worked by manipulating it. One group performed without additional help. Subjects in each of the two remaining groups could use one of two different versions of an external memory. After both exploration phases, all subjects were confronted with a recognition task ("prognostic questions", see earlier discussion) to assess their system knowledge, and with a verification task to test a representational hypothesis (see later discussion). Finally, in a third phase subjects were asked to try to reach a specified goal state as often as possible during a particular interval ("control phase"). Thereafter, subjects were again confronted with the recognition task and the verification task. The same procedure was repeated with the second automaton.

Our basic assumption is that learning about a new discrete dynamic system starts at the level of individual state transitions (S_t–I_t–S_{t+1} sequences). Later on in the course of learning these state transitions may be combined into higher-order units (see earlier). However, as a prerequisite they have to be available in working memory to become integrated. The present experiment was designed to test whether an external memory that graphically displays past states and interventions would facilitate the integration process by expanding the amount of transitions that can be made available to working memory, and whether the external memory would reduce interference between the individually experienced state transitions that share elements of the S_t–I_t–S_{t+1} triple, thereby changing subjects' exploratory behaviour. An additional question was whether the usefulness of the external memory would depend on the complexity of the system with which subjects interacted.

Hypotheses

The major focus of the experiment was on the utility of an external memory aid as one way to support identification of the unknown system structure. Existing research in this field mostly focuses on conditions that encourage people to use external memory aids spontaneously (e.g. Harris, 1980). For the most part this seems to be a question of a person's metacognitive skills and the available knowledge about the utility of an external memory aid in a particular situation. At least adults seem to know fairly well when

they should use which type of external memory aid (cf. Intons-Peterson & Fournier, 1986).

In a novel automaton, learning starts at the level of associating elements of state transitions. Particularly at the beginning of the learning process, these associations will still be fragmentary. Also, normally there will be a number of transitions that share elements of the S_t–I_t–S_{t+1} triple. An attempt to retrieve a particular transition or a component thereof will consequently be susceptible to interference from similar transitions. For instance, given a current state S_t and a desired next state S_{t+1}, an intervention I_t may be retrieved either because it is associated with the current state but not with next states different from S_{t+1} or because it is associated with the desired next state S_{t+1} but not with the current state. As a consequence, subjects may restrict their exploratory activities to a smaller number of transitions in an attempt to reduce this interference. If, in contrast, the availability of an external memory aid serves to reduce the interference, more different state transitions should be explored (*Hypothesis 1*). As a consequence of a more extensive exploratory activity, more will be learned about the structure of the automaton. Thus, subjects supported by an external memory should perform better both on the recognition task and on the control task (*Hypothesis 2*). If this is true, an external memory that in addition to preserving past information enables users to resume exploration at a part of the state transition matrix they already know could further reduce the interference and facilitate a systematic expansion of a person's individual state transition matrix, thereby causing an additional performance increase on the recognition task and on the control task (*Hypothesis 3*).

The two automata used in this experiment differed greatly in complexity as defined by formal system characteristics. Assuming that this formal property influences the learning rate, we postulate that performance should be better on the small automaton than on the complex automaton (*Hypothesis 4*). It was also of interest whether the utility of the external support system was uniform for automata of different complexities. As the external memory displays an identical number of state transitions for both automata, the relative reduction of interference should be much lower for the complex automaton. Considering the large difference in complexity between the automata (see later discussion), we postulate that the performance differences specified by Hypotheses 1 and 2 will be present for the small automaton but not for the complex automaton (*Hypothesis 5*).

A separate question concerns so-called "efficiency–divergency" effects. Oesterreich (1981) has suggested that in complex choice situations subjects prefer actions that imply more alternatives (more divergent actions) but lead to a goal less efficiently, compared to actions that are more efficient but less divergent. We wanted to see whether corresponding results could be found for dynamic task situations when subjects explore a device under conditions of imperfect knowledge. They should then prefer to explore and, consequently, acquire more knowledge about states that lead to the goal state less quickly but with a smaller risk of running into a state that is further away from the goal than the present state. In contrast, they should know less about efficient but less divergent transitions (*Hypothesis 6*).

Finally, it was intended to test a representational hypothesis that follows from the assumption about the serial learning character of associating state transitions. For that purpose, we introduced a verification task in which subjects judged whether or not a given transition was possible for the automaton they explored. If the assumption holds, we would expect faster verification times for the second of a pair of state transitions if the pair corresponded to the natural sequence of transitions in the automaton. In contrast, if the second item of the pair violated the normal sequentiality, no priming benefit should occur (*Hypothesis 7*).

The Task

Two dynamic systems were constructed on the basis of the theory of finite-state automata. They were displayed using MacFAUST.[3] MacFAUST provides a standard graphical user interface for many different kinds of discrete systems (see Figure 3).

During the exploration and control phases, subjects interacted with an automaton by clicking with the computer mouse into "input buttons" in the display. A selected button turned grey. For example, on left side of Figure 3, the combination "alpha, +,•" has been selected. For each intervention, subjects selected exactly one button in each row of input buttons and clicked "OK" when they were satisfied with their choice. Changes of selections were possible as long as "OK" had not been clicked, and "OK" was active only if one button in each row had been selected. The results of the intervention could then be observed in the "output fields" (right side of Figure 3) of the display. The input buttons were cleared, and subjects could select their next input.

The two systems used in this experiment were designed to be comparable with respect to most features and yet to differ with respect to their degree of complexity. Complexity is supposed to be a major factor influencing the difficulty of the identification task and of the utility of the external memory support. As a measure of complexity we employed

[3]MacFAUST (= Finite AUtomata Simulation Tool) runs on Apple Macintosh computers and serves to create and experiment with finite state automata as dynamic task environments. For a more detailed description of MacFAUST see Buchner, Schmitt, Funke, and Nikelowski (1991). MacFAUST is available upon request.

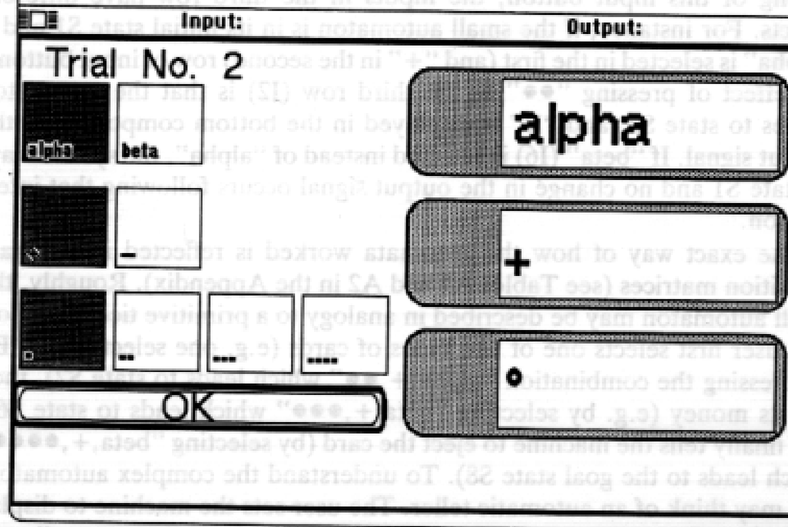


FIG. 3. Example user interface display for the small automaton. Left side: input buttons; right side: output fields.

McCabe's (1976) complexity index (see above). The small automaton has, according to this measure, a complexity of 32 (e = 39 edges and n = 9 states). The complex automaton's complexity is 260 (e = 304 edges and n = 46 states). Thus, the automata differ considerably with respect to the decisional complexity implied by their state transition matrices. The state transition matrices for both automata are given in the Appendix (Tables A1 and A2).

Both automata operate like Moore automata. The output signals reflect only the system states, and each state is associated with a unique output signal (i.e. there are no latent states). Each automaton has three rows of input buttons. In each row, one button must be selected for the complete input signal. Also, in both automata the output signal has three components (see Figure 3 and Tables A1 and A2 in the Appendix). In each automaton, one row of input buttons works similarly to an "on/off" switch (+/– and •/○ for the small and the complex automaton, respectively) in that it controls whether or not inputs in the other rows have an effect on the system in the sense that they produce a transition to a $S_{t+1} \neq S1$ state, where S1 is the initial state. For instance, if a person selected "–" in the second row of the small automaton's input buttons, the following transition always resulted in the initial state, regardless of the setting of the buttons in the remaining rows. Another row of input variables worked like a "mode" switch (alpha/beta and amount/ΔΔΔ). Depending on the current setting of this input button, the inputs in the third row have different effects. For instance, if the small automaton is in its initial state S1 and if "alpha" is selected in the first (and "+" in the second) row of input buttons, the effect of pressing "●●" in the third row (I2) is that the automaton moves to state S2, and "B" is displayed in the bottom component of the output signal. If "beta" (I6) is selected instead of "alpha", the system stays in state S1 and no change in the output signal occurs following that intervention.

The exact way of how the automata worked is reflected in the state transition matrices (see Tables A1 and A2 in the Appendix). Roughly, the small automaton may be described in analogy to a primitive ticket vendor. The user first selects one of two types of cards (e.g. one selects card "B" by pressing the combination "alpha,+,●●" which leads to state S2), then inserts money (e.g. by selecting "beta,+,●●●" which leads to state S6), and finally tells the machine to eject the card (by selecting "beta,+,●●●●" which leads to the goal state S8). To understand the complex automaton, one may think of an automatic teller. The user sets the machine to display a certain amount of money (by selecting several times "●,amount", and one of the appropriate buttons "A" to "E") and then types in one of three permissible 3-letter code words to reach the goal (by selecting several times "●,ΔΔΔ", and one of the appropriate buttons "A" to "E"). Alternatively, the user may start to type in the code and then specify the amount of money.

The automata were completely unknown to subjects, and the labels of the input and output signals were designed to be semantically poor so as to make the task primarily one of structure identification. Subjects should not use any specific knowledge they might have had about a concrete system to infer the automata's structures.

It has frequently been reported that, particularly for novices, surface features are crucial for positive transfer between different tasks (e.g. Gentner & Gentner, 1983; Novick, 1988; Schumacher & Gentner, 1988). Therefore, surface features of the two systems were made dissimilar to minimize possible transfer effects between systems. (1) The labels used for the input and output variables and for their levels were changed from one system to the other. (2) The spatial positions of the input buttons and the output fields on the graphical display were different for both automata.

The External Memory

For two of the three experimental groups, an external memory was available during the exploration phase. Two different versions of the external memory were implemented. Both versions graphically displayed six past states at a time in a separate window, and the window automatically appeared on the screen every six interventions. The display consisted of

scaled-down copies of the original displays and showed the selected input buttons, together with the following output signal. An example of the external memory for the small automaton is presented in Figure 4. Subjects could inspect all past transitions of the current exploration phase by clicking into the numbered top row of the window. The state with the appropriate ordinal number and its five predecessors were then displayed in the memory window. Subjects clicked into the window's "close box" in the top left corner when they wanted to continue to explore the automaton. This "static" version of the external memory was presented to one group. For another group the external memory additionally included the "dynamic" option to make one of the old states displayed in the memory window the next present state of the system and to continue system exploration at that state. To achieve this, subjects clicked into the part of the window displaying the desired next state. They were then asked to confirm their decision before the memory window was closed. The automaton then displayed the desired next state as its current state from which subjects could continue their exploration. The transition was counted like a normal user intervention.

Knowledge Assessment

After each interval of 50 interventions, subjects were confronted with a recognition test and a verification task. The recognition test consisted of 10 items, the verification task consisted of 20 items.

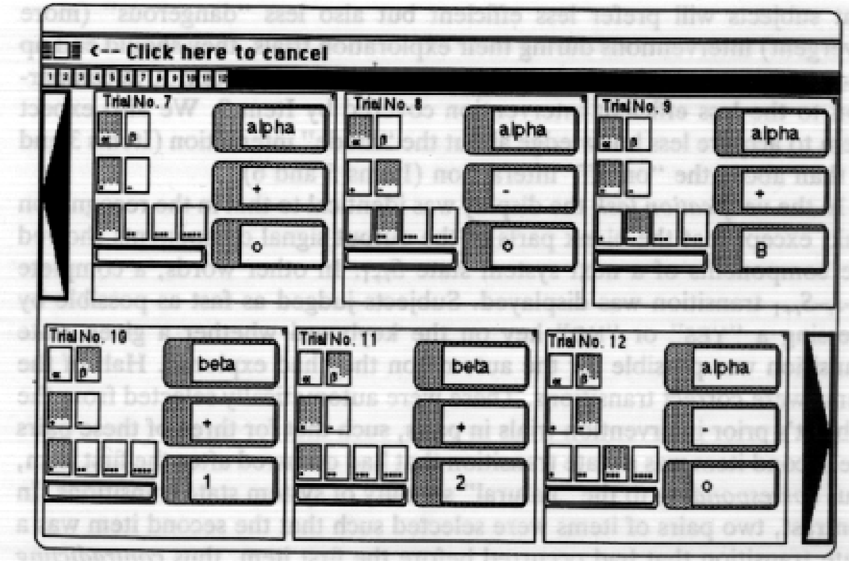


FIG. 4. Example display of the external memory for the small automaton. All trials have ordinal numbers, with that of the present trial being the highest.

During the *recognition test* the screen display was identical to the display during the intervention trials, except that the three output signal displays were divided horizontally into two separate fields, one of which showed a system state S_t and the other part was empty. Also, three input buttons were shaded grey to indicate an intervention I_t. Subjects' task was to consider the state S_t and the intervention I_t, and then to select from a list of possible and an equal number of impossible output signals the appropriate signal of the next state S_{t+1} ("prognostic question", see earlier discussion). More specifically, for each possible output signal in one of the three components there was one impossible alternative. Subjects selected the three components by clicking into the output fields. Each click in one of the fields brought up a different output signal component. Subjects clicked "OK" when they thought the displayed components constituted the correct output signal.

A complete list of the recognition items for both automata is given in Table A3 in the Appendix. The items did not represent a random sample from the state transition matrix but were selected to cover certain interesting features of the automaton. For instance, Items 9 and 10 cover inputs with a different efficiency–divergency characteristic (see Hypothesis 6). To illustrate, consider the complex automaton in which two I3 interventions may replace seven I1 interventions to reach state S35 from the initial state S1. On the other side, for intervention I3 there is a higher risk of ending up in State S12, from which the distance to the Goal State S45 is maximal (only resets to the initial state are possible). Thus, as we expect that subjects will prefer less efficient but also less "dangerous" (more divergent) interventions during their exploration trials, they should end up knowing less about the efficient intervention covered by Item 10 in comparison to the less efficient intervention covered by Item 9. We also expect them to acquire less knowledge about the "mode" interaction (Items 3 and 4) than about the "on/off" interaction (Items 5 and 6).

In the *verification task* the display was identical to that in the recognition task, except that the blank parts of the output signal components showed the components of a next system state S_{t+1}. In other words, a complete S_t–I_t–S_{t+1} transition was displayed. Subjects judged as fast as possible by pressing a "YES" or "NO" key on the keyboard whether a given state transition was possible for the automaton they had explored. Half of the items were correct transitions. These were automatically selected from the subject's prior intervention trials in pairs, such that for three of these pairs the second item was a state transition that had occurred after the first item, thus *corresponding* to the "natural" seriality of system state transitions. In contrast, two pairs of items were selected such that the second item was a state transition that had occurred before the first item, thus *contradicting* the normal seriality of state transitions.

During the *control task* subjects interacted with the automaton as during the exploration phase, but this time the instruction was to use the shortest possible sequence of interventions to reach the goal state (States S8 and S45 for the small and the complex automaton, respectively) as often as possible within 50 interventions. For the small automaton the optimal sequence involves three, for the complex automaton it involves six interventions. Every time the goal state is reached, an additional transition is used for the reset to the automaton's initial state. The external memory was not available during the control phase.

A final dependent measure was taken directly from subjects' exploration trials. It was counted how many *different state transitions* subjects generated while they attempted to learn how the automaton operated.

Method

Subjects. Subjects were 68 Bonn University students who either volunteered or participated to fulfil course requirements. They were aged 20 to 40 years.

Design. Subjects were randomly assigned to one of the three experimental conditions, these being no memory (NM), static memory (SM), and dynamic memory (DM). There were 23 subjects in Groups NM and SM, and 22 subjects in Group DM. Because subjects performed on two successive automata, one half within each group started with the complex automaton, the other half started with the small automaton.

Procedure. Subjects were tested individually. The instructions were read to them in a standardized form and repeated on the computer. All subjects received a printed version of the graphical display of the system and a description of the course of the experiment. They were instructed that they would be confronted with two unknown automata and that their task was to identify how these automata operated. They were told that each automaton had one particular goal state and that if they would reach this state, the automaton would present a brief signal indicating their "success" and would then automatically reset itself to the initial state from where they could then resume exploration. Subjects in the memory conditions were informed about the external memory and instructed how to handle the memory window. Subjects performed about 30 interventions on an extremely simple "learning automaton" to become acquainted with the use of the computer mouse.

Each subject then performed on two successive automata. For each automaton they carried out two exploration phases and one control phase. Each of the exploration phases and the control phase consisted of 50 intervention trials. During the two exploration phases, subjects' task was to learn how the automaton operated by manipulating it and observing the state changes. For the memory conditions, the memory window was displayed automatically every six trials during the two exploration phases.

In contrast, during the control phase subjects were instructed to try to reach the goal state as often as possible. None of the groups received external support while performing on this task.

After each exploration phase and after the control phase subjects performed on the recognition task (10 items) and on the verification task (20 items). The order of presentation of single items was randomized for the recognition task. For the verification task, the order of presentation of *pairs* of items (presented in contradicting or corresponding sequence) was randomized for each subject. After each recognition item, subjects were asked to indicate the degree to which they felt their choice was appropriate, on a scale from 1 (= guess) to 4 (= perfect confidence).

The same procedure was repeated with the second automaton, except that subjects were not specifically instructed for the second automaton and simply told that their task and the procedure were the same as before but that the automaton was different.

Results

A multivariate approach was used to analyse the repeated measures data statistically (O'Brian & Kaiser, 1985). The Pillai-Bartlett-*V* criterion, known for its robustness, was chosen as the multivariate test statistic (Olson, 1976). However, the *F*-approximation to the distribution of *V* is used in describing the results. For all analyses, the critical level of α and β was set to 0.05, which is sufficient to detect "large effects" given $N = 23$ in each of the three experimental groups. Also, for every significant effect partial R^2s (R_p^2) will be reported as a measure of the proportion of variance explained relative to the total variance not explained by other experimental variables (Cohen, 1977).

We first analysed whether the availability of an external memory had an influence on subjects' exploratory activities by reducing the interference from similar state transitions (Hypothesis 1). If this was the case, subjects in Groups SM and DM should have exposed themselves to more different state transitions than subjects in Group NM. Table 1 (upper section) displays the relevant data. Two ANOVAs were run separately for the two automata with planned contrasts to compare the memory groups to Group NM. The *F*-tests yielded significant group differences only for the small automaton, $F(2, 65) = 3.93$, $R_p^2 = 0.11$ [vs. $F(2, 65) < 1.38$ for the complex automaton]; a direct comparison showed that the difference in the small automaton is due to Group NM's lower number of different state transitions relative to the memory groups [$t(65) = -2.56$], whereas there is no difference between Groups SM and DM [$t(65) < 1.14$].

TABLE 1
Mean Number of Different State Transitions and "Off" States during the Exploration Phases for the Three Experimental Groups and the Two Different Automata

		Groups		
Variable	*Automaton*	*NM*	*SM*	*DM*
Different state transitions	Small	33.9 (9.1)	38.5 (8.6)	41.6 (10.0)
	Complex	60.9 (17.1)	66.8 (14.2)	67.3 (12.3)
"Off" states	Small	56.9 (17.2)	45.9 (16.0)	40.9 (16.0)
	Complex	40.7 (15.8)	35.8 (19.9)	32.1 (18.4)

Note: Standard deviations are given in parentheses.

Table 1 also displays how many trials subjects spent with the automaton "switched off", which we analysed for exploratory purposes. In contrast to any of the other dependent variables considered here, this represents a possible system-specific measure of performance. A person who is better at acquiring the "on/off" concept should "waste" fewer intervention trials with the automaton switched off. For the small automaton, we find overall group differences, $F(2, 65) = 5.53$, $R_p^2 = 0.15$; planned contrasts revealed significant differences between the memory groups and Group NM, $t(65) = 3.17$, but not between Groups SM and DM [$t(65) < -1.03$]. For the complex automaton, the overall test indicated that the group means do not differ significantly [$F(2, 65) < 1.28$].

The data from the control task are shown in Table 2. An ANOVA with group as independent factor showed that there is no significant difference for either the small automaton [$F(2, 65) < 1.79$] or the complex automaton [$F < 1$] in terms of how often subjects reached the goal state during the control task. Considering, however, the low overall number of goal states reached and the fact that the means are in the expected direction at a descriptive level, one might suspect a floor effect. If one applies the optimal sequence of interventions for the small automaton, the goal state can be reached 12 times during 50 intervention trials (three transitions from the Initial State S1 to S8 plus one "autonomous" transition back to the initial state). With the complex automaton the goal state can be reached five times at the most. It is possible that the automata were too difficult to control, given the number of exploration trials.

The knowledge acquired about the automaton should be reflected in the number of correct responses on the recognition test (see Figure 5). A

TABLE 2
Mean Number of Goal States Reached during the Control Phase for the Three Experimental Groups and the Two Different Automata

	Groups		
Automaton	*NM*	*SM*	*DM*
Small	0.7 (2.4)	2.6 (4.3)	1.6 (3.3)
Complex	0.4 (1.3)	0.9 (1.8)	0.7 (1.8)

Note: Standard deviations are given in parentheses.

global F-test indicates performance differences on the small automaton between groups, $F(2, 65) = 7.23$, $R_p^2 = 0.18$; planned contrasts show that whereas there is no difference between Groups SM and DM [$t(65) < 1.08$], the difference between the memory groups and Group NM is significant, $t(65) = -3.65$. The memory groups have more knowledge available about the automaton than does the group without support. For the complex automaton, the global F-test yields no significant group differences ($F < 1.11$). It is interesting to note that the confidence ratings that followed each recognition item show exactly the same pattern. For the small auto-

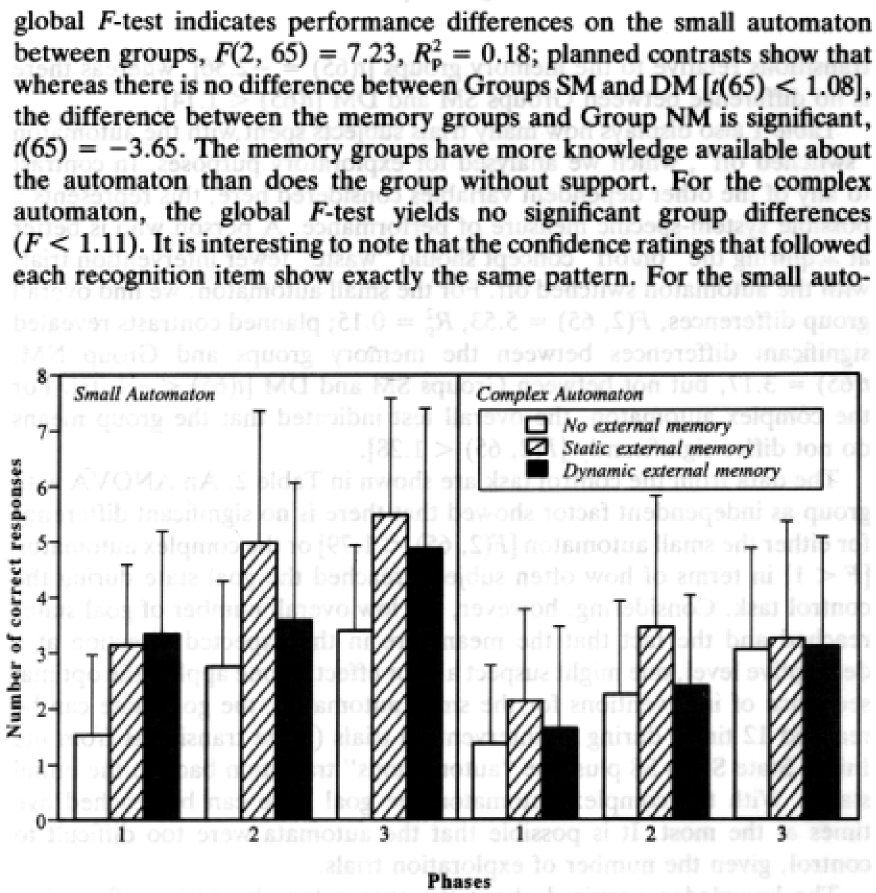


FIG. 5. Average number of correct responses in the recognition tasks following the first (1) and second (2) exploration phase and the control phase (3) for the small and the complex automaton. The error bars represent the standard deviations.

maton, there are significant group differences, $F(2, 65) = 6.41$, $R_p^2 = 0.17$; the difference exists between the memory groups and Group NM, $t(65) = 3.37$, and there is no difference between Groups SM and DM [$t(65) < 1.23$]. For the complex automaton, there are no significant group differences [$F(2, 65) < 1.78$].

Considering performance on the small automaton, the data seem to be compatible with Hypotheses 1 and 2 (with the exception of the control phase). Performance benefits due to the availability of an external memory can be observed. However, Groups SM and DM consistently do not show any performance differences. Thus, Hypothesis 3 can be rejected.

In general, performance is worse on the complex automaton for both the number of goal states reached [$F(1, 65) = 6.33$, $R_p^2 = 0.09$] and the number of correct responses to recognition items [$F(1, 65) = 40.99$, $R_p^2 = 0.39$]. This pattern has been specified by Hypothesis 4. Together with the fact that performance benefits for the memory groups occurred only for the small automaton, this points to the differential utility of the external memory support. It was effective only when subjects interacted with the less complex of two otherwise very similar automata (Hypothesis 5).

The recognition test also illustrates that subjects continuously acquire more knowledge about the automata. A MANOVA with phases as within-subjects factor and planned contrasts confirms that the number of correct responses on the recognition task increases monotonically as a function of the number of trials on the task, $F(2, 64) = 64.13$, $R_p^2 = 0.67$, from the first to the second exploration phase, $F(1, 65) = 40.35$, and from the first two exploration phases to the subsequent control phase, $F(1, 65) = 123.58$.

A more detailed analysis of some of the items of the recognition test revealed that efficiency–divergency effects (Hypothesis 6) were found for the complex automaton, $F(1, 65) = 10.40$, $R_p^2 = 0.14$, but not for the small automaton [$F(1, 65) < 1.83$]. This may be due to the difference in efficiency as implemented in the automata. The difference is larger for the critical items in the complex automaton (see Table A2).

Table 3 also shows the "mode" interaction (items 3 and 4, see Table A3) was more difficult to understand than the "on/off" interaction (items 5 and 6) of the input variables (see Table 3). A MANOVA with item type as within-subjects factor revealed significant differences between the two different types of items, $F(1, 65) = 60.91$, $R_p^2 = 0.48$.

Finally, our concern was whether we would find the representational effects specified by Hypothesis 7 in the verification task. If the seriality of events (i.e. state transitions) as experienced during exploration is mentally represented, then reactions to the second member of a pair of items "corresponding" to this seriality should be facilitated. In contrast, reaction times to "contradicting" items and to the first members of the pair should be slower. The latter items may be called "neutral" because they follow a distractor item. Only reaction times of correct answers were entered into

TABLE 2
Mean Number of Goal States Reached during the Control Phase for the Three Experimental Groups and the Two Different Automata

	Groups		
Automaton	*NM*	*SM*	*DM*
Small	0.7 (2.4)	2.6 (4.3)	1.6 (3.3)
Complex	0.4 (1.3)	0.9 (1.8)	0.7 (1.8)

Note: Standard deviations are given in parentheses.

global *F*-test indicates performance differences on the small automaton between groups, $F(2, 65) = 7.23$, $R_p^2 = 0.18$; planned contrasts show that whereas there is no difference between Groups SM and DM [$t(65) < 1.08$], the difference between the memory groups and Group NM is significant, $t(65) = -3.65$. The memory groups have more knowledge available about the automaton than does the group without support. For the complex automaton, the global *F*-test yields no significant group differences ($F < 1.11$). It is interesting to note that the confidence ratings that followed each recognition item show exactly the same pattern. For the small auto-

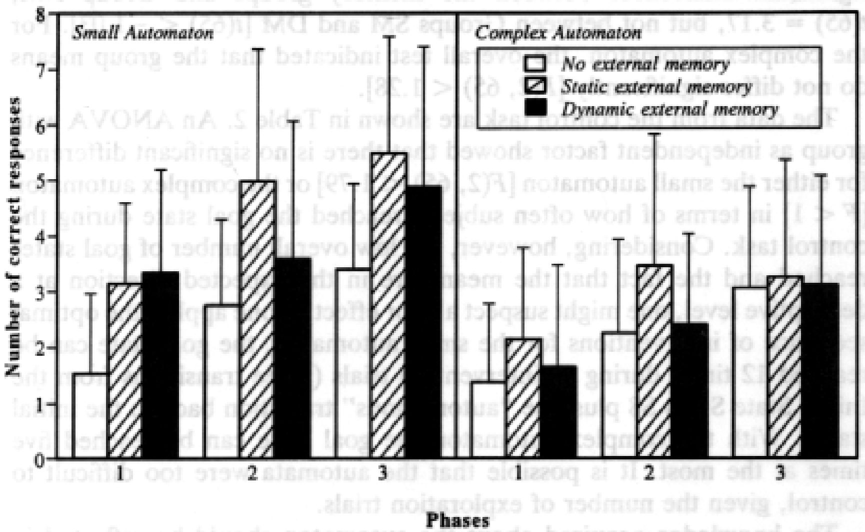


FIG. 5. Average number of correct responses in the recognition tasks following the first (1) and second (2) exploration phase and the control phase (3) for the small and the complex automaton. The error bars represent the standard deviations.

maton, there are significant group differences, $F(2, 65) = 6.41$, $R_p^2 = 0.17$; the difference exists between the memory groups and Group NM, $t(65) = 3.37$, and there is no difference between Groups SM and DM [$t(65) < 1.23$]. For the complex automaton, there are no significant group differences [$F(2, 65) < 1.78$].

Considering performance on the small automaton, the data seem to be compatible with Hypotheses 1 and 2 (with the exception of the control phase). Performance benefits due to the availability of an external memory can be observed. However, Groups SM and DM consistently do not show any performance differences. Thus, Hypothesis 3 can be rejected.

In general, performance is worse on the complex automaton for both the number of goal states reached [$F(1, 65) = 6.33$, $R_p^2 = 0.09$] and the number of correct responses to recognition items [$F(1, 65) = 40.99$, $R_p^2 = 0.39$]. This pattern has been specified by Hypothesis 4. Together with the fact that performance benefits for the memory groups occurred only for the small automaton, this points to the differential utility of the external memory support. It was effective only when subjects interacted with the less complex of two otherwise very similar automata (Hypothesis 5).

The recognition test also illustrates that subjects continuously acquire more knowledge about the automata. A MANOVA with phases as within-subjects factor and planned contrasts confirms that the number of correct responses on the recognition task increases monotonically as a function of the number of trials on the task, $F(2, 64) = 64.13$, $R_p^2 = 0.67$, from the first to the second exploration phase, $F(1, 65) = 40.35$, and from the first two exploration phases to the subsequent control phase, $F(1, 65) = 123.58$.

A more detailed analysis of some of the items of the recognition test revealed that efficiency–divergency effects (Hypothesis 6) were found for the complex automaton, $F(1, 65) = 10.40$, $R_p^2 = 0.14$, but not for the small automaton [$F(1, 65) < 1.83$]. This may be due to the difference in efficiency as implemented in the automata. The difference is larger for the critical items in the complex automaton (see Table A2).

Table 3 also shows the "mode" interaction (items 3 and 4, see Table A3) was more difficult to understand than the "on/off" interaction (items 5 and 6) of the input variables (see Table 3). A MANOVA with item type as within-subjects factor revealed significant differences between the two different types of items, $F(1, 65) = 60.91$, $R_p^2 = 0.48$.

Finally, our concern was whether we would find the representational effects specified by Hypothesis 7 in the verification task. If the seriality of events (i.e. state transitions) as experienced during exploration is mentally represented, then reactions to the second member of a pair of items "corresponding" to this seriality should be facilitated. In contrast, reaction times to "contradicting" items and to the first members of the pair should be slower. The latter items may be called "neutral" because they follow a distractor item. Only reaction times of correct answers were entered into

TABLE 3
Average Number of Correct Answers to an Item on the Recognition Task Following the Two Exploration Phases and the Control Phase

	Items 3 & 4 "mode" interaction	*Items 5 & 6 "on/off" interaction*	*Item 9 "less efficient input"*	*Item 10 "efficient input"*
Small Automaton			0.54 ←n.s.→	0.69
Complex Automaton	0.83 ←s.→	1.51	1.13 ←s.→	0.75

Note: Maximum is three correct answers.

the analyses.[4] Reaction times to contradicting items are significantly slower than reaction times to corresponding items [see Figure 6: $F(2, 57) = 4.51$, $R_p^2 = 0.14$, and $F(2, 54) = 4.95$, $R_p^2 = 0.16$, for the small and complex automaton, respectively]. In contrast, reaction times to contradicting items are not different from reaction times to neutral items (both $Fs < 1$). Presenting items in their normal order facilitates their verification. These data are in line with Hypothesis 7.

Discussion

The most prevalent result of the present experiment is that the external memory support showed beneficial effects on a number of different performance measures, but only for the less complex of two automata. It is suggested that this finding is best interpreted in analogy to the differential approach known from clinical psychology according to which we may ask

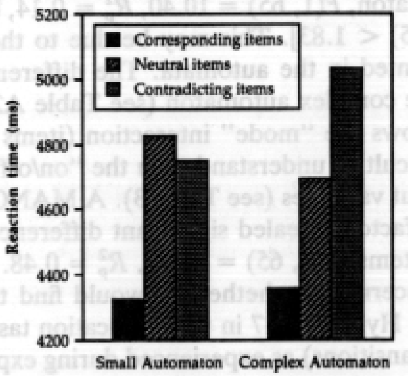


FIG. 6. Mean reaction times for correct responses to different types of items in the verification tasks.

[4]Note that due to the limited number of such items some subjects had to be excluded from the analyses because of missing values. As a result, the degrees of freedom for these analyses deviate from what would be normal for the entire sample.

which type of support system is indicated in which situation under which type of task demand for which type of person. Consequently, a desirable strategy of support system design would take into account as much as possible person variables, situational variables, and task variables. In the present case, complexity as a task variable had a differential effect on the utility of the support system. Subjects benefited from the external memory only when they explored the small automaton.

The external support did not affect performance on the number of goal states subjects reached during the control phase. This variable seems to represent a difficult aspect of the experimental task, and the results could reflect a floor effect. In this case giving subjects more trials to explore the system should reveal group differences. Alternatively, one could develop more refined measures of control performance. How often subjects reach a certain goal state is only a very global assessment of their control performance. It seems possible to design tasks at different levels of difficulty that are indicative of subjects' knowledge state (cf. Falmagne, 1989).

In no case was the performance of Group DM better than that of Group SM. This finding runs contrary to our expectations. We therefore analysed how often subjects in Group DM actually used the option to make a well-known old state the new state from which to resume exploration. It turned out that just about half of the subjects in Group DM (12 out of 23) ever used the option. Those subjects who used it did so on an average of only 4.1 (out of a maximum of 16) occasions. This could explain why Group DM did not perform better than Group SM. The external memory itself was automatically displayed in a fixed interval of six trials. In contrast, the "dynamic" option was left at subjects' disposal. More importantly, there was no obvious visual reminder of the option on the screen (subjects simply clicked onto the desired state display in the memory window—see Figure 4). It might be that such design factors contributed to the fact that the option was mostly ignored.

With respect to the number of different state transitions, it is interesting to note that the memory manipulation in this case affected a variable that reflects a qualitative aspect of subjects' exploratory behaviour. Our interpretation of this finding is that the availability of past system states on an external medium reduces the interference in memory that is otherwise caused by state transitions that share identical elements of the S_i–I_i–S_{i+1} triple.

The efficiency–divergency effect had been demonstrated for the complex automaton only. This could be due to the somewhat smaller efficiency–divergency difference between the critical inputs in the small automaton. If the interpretation is correct, the finding indicates that exploring an unknown device parallels a complex choice situation in that subjects prefer actions that imply more alternatives but lead to a goal less efficiently compared to actions that are more efficient but less divergent.

Reaction times to state transitions in the verification task yielded useful information about how discrete dynamic systems might be represented. If the second item of a pair was a state transition that in the chronology of system events occurred after the first item, reaction times were faster than when the second item had actually occurred before the first member of the pair. The facilitative effect indicates that seriality is an important aspect of how experience with a dynamic system is mentally represented. This could have implications for the development of new diagnostic procedures to assess system knowledge more adequately. In the present experiment, the items in the diagnostic phases were independent, and their sequence was randomized. In addition, they were static, presenting S_t, I_t, and the new (to be recognized) S_{t+1} at the same time. Thus, one possible consequence for the further development of diagnostic procedures could be to probe subjects' knowledge with *sequences* of items that correspond to the chronology of system events.

GENERAL DISCUSSION

The purpose of this paper was to illustrate how a well-elaborated theoretical background such as the theory of finite-state automata can provide the basis for an interesting experimental paradigm in research on knowledge acquisition and knowledge application using computer-simulated "microworlds". More specifically, the approach can be helpful in constructing classes of task environments that can be compared with respect to formal characteristics, and it facilitates the direct manipulation of such characteristics by providing a common formal background. We have also argued that the formalism for describing automata can be used to select plausible psychological hypotheses about, for instance, how learning-by-exploration may proceed, and how system information is represented.

Another interesting aspect of the approach is that it directly suggests a number of methods to assess a person's knowledge about a system. These methods are both general in the sense that they can be applied to any automaton (e.g. "prognostic questions"), and related to "classical" measures of memory like cued recall, recognition, or verification tasks. Such items may be either randomly sampled from the state transition matrix, or they may be selected to cover critical features of the task environment. The latter procedure has been used in the recognition task in the present experiment. With respect to the verification task, we would like to point out that methods of "mental chronometry" (Posner, 1978), which have up to now been absent in this research domain, were shown to be applicable to assessing the way system information is represented in memory.

In addition, finite-state automata theory provides a criterion for optimal performance that is essential for evaluating a person's system control behaviour. However, other performance measures are also readily available. For instance, we have attempted to demonstrate that the number of different state transitions a person decides to explore can be an interesting dependent variable. Another example could be to compare recognition items covering transitions that have actually been explored to items covering transitions that have not. This procedure may turn out to be useful for assessing generalization processes such as a person's discovery of an "ON/OFF" or "mode" interaction. To conclude this point, we would like to argue that these more general methods of knowledge assessment are a rather attractive alternative to idiosyncratic diagnostic procedures that are directly derived from task parameters such as "production output" (e.g. Dörner, 1987; Morris & Rouse, 1985).

We also wish to emphasize the parallels between finite-state automata and finite grammars (e.g. Chomsky & Miller, 1958). Finite grammars *generate* structured event sequences such as sentences over an alphabet, whereas finite state automata "*understand*" these sequences. This suggests a close relationship between tasks that involve identifying the structure of an automaton and tasks that require subjects to process material that was generated by a finite grammar (e.g. Brooks, 1978; Dulany, Carlson, & Dewey, 1984; Reber, 1989).

Of course, using a finite-state automata framework in experimental psychology is not an entirely new idea. Suppes (1969), for instance, suggested the use of automata theory as a theoretical background for modelling animal and human behaviour. Another application of automata theory is the construction of machine prototypes and the analysis of human effort to reach certain goals. Such analyses were done by Bösser and Melchior (1992). However, the potential of automata theory as a framework for generating classes of dynamic task environments and appropriate diagnostic instruments has to our knowledge not been outlined before.

The most serious problem of the finite state automata approach to human interaction with dynamic systems probably is one of system complexity. In principle, it is possible to design discrete dynamic systems of any complexity.[5] However, there is a practical upper limit in terms of computer memory and computational effort as well as in terms of the effort it takes the experimenter to construct very large state transition matrices. For instance, the simulation of a complicated industrial production process clearly is beyond the capability of the approach. This limitation certainly reduces the applicability of the discrete systems framework.

We can see two partial remedies to the problem: (1) It may not be necessary to simulate the entire complexity of a learning environment to investigate the basic cognitive processes involved in interacting with it.

[5]It should be noted that most so-called "continuous" systems are *in principle* discrete in that they accept as input and display as output only discrete numbers. On the other hand, a discrete systems' output may appear continuous if it has a sufficiently large number of levels.

Depending on the system, it could be possible to omit certain details or simulate only parts of the system as long as "simulation fidelity" (Hays & Singer, 1989) can be preserved. (2) A second aspect is that some of the advantageous properties of the approach, such as the diagnostic procedures and representational assumptions discussed above, can be utilized as long as one *conceives* of the system in terms of a finite-state automaton. The actual implementation may be quite different.

Yet another problem common to many approaches to human interaction with complex dynamic task environments is finding the proper level of abstraction for characterizing a device. For the case of learning about an entirely new device, we have assumed that different states, interventions, and next states must become associated. In this situation learning occurs on the level of physical events in the automaton. After some experience, however, individual state transitions may be combined into chunks. Such chunks may be "horizontal" combinations of transitions into routines to get a system reliably from one state to a distant state, or they may be "vertical" combinations across, for example, a specific intervention like an "ON/OFF" switch. The problem of developing an adequate model for such abstraction processes is aggravated by the fact that concepts may be available at different degrees of abstraction. Concepts at different degrees of abstraction enable more experienced learners to attend to (and, hence, control) a task at different levels. Naturally, if a person has prolonged experience with attending to a task at a high level of abstraction, it may be difficult and effortful to recollect individual states and interventions when required to do so, either outside the context of the immediate control process or if an unexpected event, like a failure, interrupts an ongoing procedure. This is assumed because normally what we are able to recall is information we actively attend to (Kellog, 1980; Norman, 1969). Moray (1990), for instance, has presented a "lattice theory" designed to describe such abstraction phenomena occurring at different levels. He does not, however, make any assumptions about cognitive processes involved in abstraction, nor does he specify how shifts between different levels of abstractions might occur.

The processes underlying abstractions of the type described here will certainly need more attention in the future. Nevertheless it can be interesting to analyse learning processes at the elementary level of state transitions. In another study (Buchner & Funke, 1991), we investigated early learning in and transfer of associations between finite-state automata. After initially acquiring knowledge about a "source automaton" (a simplified radio with a built-in alarm device), several groups of subjects performed on different "target automata". The state transition matrices underlying the target automata were identical for all groups but completely different from that of the source automaton. The automata differed with respect to the labelling of their input and output signals. In one condition these labels were entirely new, whereas in a different condition the original labels had been preserved. If learning indeed occurs at the level of associating states, interventions, and next states, the latter case should correspond to the A–B, A–Br situation in paired associate learning (cf. Martin, 1965). Stimuli and responses from the first list are preserved in the transfer list except that they are repaired. This is known to produce considerable negative transfer. In accordance with our expectations, it was observed that those subjects performed worst that interacted with the target automaton in which the originally associated elements were preserved but were also repaired (due to an entirely different transition matrix).

In summary, there are definite problems and limitations that come with the approach we advocated for in this paper. However, we believe that its attractive features make it worth exploring further the applicability to research using dynamic task environments. In this sense the finite-state automata approach could provide an interesting additional way of addressing problems of human knowledge acquisition and knowledge use in dynamic tasks, and it could have the potential to stimulate research in the area.

REFERENCES

Anderson, J.R. (1982). The acquisition of cognitive skill. *Psychological Review, 89*, 369–406.

Ashby, W.R. (1956). *An introduction to cybernetics*. London: Chapman & Hall.

Berry, D.C., & Broadbent, D.E. (1984). On the relationship between task performance and associated verbalizable knowledge. *The Quarterly Journal of Experimental Psychology, 36A*, 209–231.

Berry, D.C., & Broadbent, D.E. (1987). The combination of explicit and implicit learning processes in task control. *Psychological Research, 49*, 7–15.

Berry, D.C., & Broadbent, D.E. (1988). Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology, 79*, 251–272.

Bösser, T., & Melchior, E.-M. (1992). The SANE toolkit for cognitive modelling and user centred design. In M. Galer, S. Harker, & J. Ziegler (Eds.), *Methods and tools in user-centered design for information technology* (pp. 93–126). Amsterdam: North Holland.

Brehmer, B. (1987). Development of mental models for decision in technological systems. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New technology and human error* (pp. 111–120). Chichester: John Wiley & Sons.

Brehmer, B., Leplat, J., & Rasmussen, J. (1991). Use of simulation in the study of complex decision making. In J. Rasmussen, B. Brehmer, & J. Leplat (Eds.), *Distributed decision making: Cognitive models for cooperative work* (pp. 373–386). New York: Wiley.

Broadbent, D.E., Berry, D.C., & Gardner, P. (1990). *The role of action and of advice in knowledge acquisition* (KAUDYTE Report 2PR1DEB). Oxford: University of Oxford.

Broadbent, D.E., Fitzgerald, P., & Broadbent, M.H.P. (1989). Implicit and explicit knowledge in the control of complex systems. *British Journal of Psychology, 77*, 33–50.

Brooks, L. (1978). Non-analytic concept formation and memory for instances. In E. Rosch & B.B. Lloyd (Eds.), *Cognition and categorization* (pp. 169–211). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Buchner, A. (1993). *Implizites Lernen: Probleme und Perspektiven* [Implicit learning: Problems and future perspectives]. Weinheim, FRG: Psychologie Verlags Union.
Buchner, A., & Funke, J. (1991). Transfer of associations in finite state automata. *Berichte aus dem Psychologischen Institut der Universität Bonn, 17(2)*.
Buchner, A., Schmitt, L., Funke, J., & Nikelowski, B. (1991). MacFAUST: Program for constructing finite automata as instruments in problem solving research. Manual. *Berichte aus dem Psychologischen Institut der Universität Bonn, 17* (3).
Chomsky, N., & Miller, G.A. (1958). Finite state languages. *Information and Control, 1*, 91–112.
Cohen, J. (1977). *Statistical power analysis for the behavioral sciences*. Revised edition. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
Dörner, D. (1987). On the difficulties people have in dealing with complexity. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New technology and human error* (pp. 97–109). Chichester: John Wiley & Sons.
Dulany, D.E., Carlson, R.A., & Dewey, G.I. (1984). A case of syntactical learning and judgement: How conscious and how abstract? *Journal of Experimental Psychology: General, 113*, 541–555.
Falmagne, J.-C. (1989). A latent trait theory via a stochastic learning theory for a knowledge space. *Psychometrika, 54*, 283–303.
Frensch, P.A. (1991). Transfer of composed knowledge in a multi-step serial task. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17*, 997–1016.
Funke, J. (1988). Using simulation to study complex problem solving: A review of studies in the FRG. *Simulation & Games, 19*, 277–303.
Funke, J. (1990). Systemmerkmale als Determinanten des Umgangs mit dynamischen Systemen [System features as determinants of dynamic task environments]. *Sprache & Kognition, 9*, 143–154.
Funke, J. (1991). Solving complex problems: Exploration and control of complex systems. In R.J. Sternberg & P.A. Frensch (Eds.), *Complex problem solving: Principles and mechanisms* (pp. 185–222). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
Gentner, D., & Gentner, D.R. (1983). Flowing waters or teeming crowds: Mental models of electricity. In D. Gentner & A.L. Stevens (Eds.), *Mental models* (pp. 99–129). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
Gillund, G., & Shiffrin, R. (1984). A retrieval model of both recognition and recall. *Psychological Review, 19*, 1–65.
Harcum, E.R. (1953). Verbal learning of overlearned forward and backward associations. *American Journal of Psychology, 66*, 622–625.
Harris, J.F. (1980). Memory aids people use: Two interview studies. *Memory & Cognition, 8*, 31–38.
Hayes, N.A., & Broadbent, D.E. (1988). Two modes of learning for interactive tasks. *Cognition, 28*, 249–276.
Hays, R.T., & Singer, M.J. (1989). *Simulation fidelity in training system design. Bridging the gap between reality and training*. New York: Springer.
Hesse, F.W. (1982). Effekte des semantischen Kontexts auf die Bearbeitung komplexer Probleme [Effects of semantic context on complex problem solving]. *Zeitschrift für Experimentelle und Angewandte Psychologie, 29*, 62–91.
Hoc, J.-M. (1989). Cognitive approaches to process control. In G. Tiberghien (Ed.), *Advances in cognitive science. Volume 2: Theory and applications* (pp. 178–202). Chichester: Ellis Horwood.
Hopcroft, J.E., & Ullmann, S.D. (1979). *Introduction to automata theory, languages, and computation*. Reading, MA: Addison-Wesley.
Hunt, R.M., & Rouse, W.B. (1981). Problem solving skills of maintenance trainees in diagnosing faults in simulated power plants. *Human Factors, 23*, 317–328.

Intons-Peterson, M.J., & Fournier, J. (1986). External and internal memory aids: When and how often do we use them? *Journal of Experimental Psychology: General, 115*, 267–280.
Jeffries, R., Polson, P.G., & Razran, L. (1977). A process model for missionaries—cannibals and other river-crossing problems. *Cognitive Psychology, 9*, 412–440.
Kellog, R.T. (1980). Is conscious attention necessary for long-term storage? *Journal of Experimental Psychology: Human Learning and Memory, 6*, 379–390.
Kotovsky, K., & Fallside, D. (1989). Representation and transfer in problem solving. In D. Klahr & K. Kotovsky (Eds.), *Complex information processes* (pp. 69–108). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
MacKay, D.G. (1982). The problems of flexibility, fluency, and speed-accuracy in skilled behavior. *Psychological Review, 89*, 483–506.
Martin, E. (1965). Transfer of verbal paired associates. *Psychological Review, 72*, 327–343.
McCabe, T.J. (1976). A complexity measure. *IEEE Transactions on Software Engineering, SE-2*, 308–320.
Moray, N. (1990). A lattice theory approach to the structure of mental models. In D.E. Broadbent, A. Baddeley, & J.J. Reason (Eds.), *Human factors in hazardous situations* (pp. 577–583). Oxford: Clarendon Press.
Moray, N., Lootsteen, P., & Pajak, J. (1986). Acquisition of process control skills. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-16*, 497–504.
Morris, N.M., & Rouse, W.B. (1985). The effects of type of knowledge upon human problem solving in a process control task. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-15*, 698–707.
Murdock, B.B. (1956). "Backward" learning in paired associates. *Journal of Experimental Psychology, 51*, 213–215.
Murdock, B.B. (1958). "Backward" associations in transfer and learning. *Journal of Experimental Psychology, 55*, 111–114.
Newell, A., & Rosenbloom, P.S. (1981). Mechanisms of skill acquisition and the law of practice. In J.R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 1–56). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
Norman, D.A. (1969). Memory while shadowing. *The Quarterly Journal of Experimental Psychology, 21*, 85–93.
Novick, L.R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 510–520.
O'Brian, R.G., & Kaiser, M.K. (1985). MANOVA method for analyzing repeated measures designs: An extensive primer. *Psychological Bulletin, 97*, 316–333.
Oesterreich, R. (1981). *Handlungsregulation und Kontrolle* [Action regulation and control]. München, FRG: Urban & Schwarzenberg.
Olson, C.L. (1976). On choosing a test statistic in multivariate analysis of variance. *Psychological Bulletin, 83*, 579–586.
Plötzner, R., Spada, H., Stumpf, M., & Opwis, K. (1990). Learning qualitative and quantitative reasoning in a microworld for elastic impacts. *European Journal of Psychology of Education, 4*, 501–516.
Posner, M.I. (1978). *Chronometric explorations of mind*. Hillsdale, NJ: Lawrence Erlbaum Associates. Inc.
Putz-Osterloh, W. (1990). *Experiment on system control and transfer to deviations* (KAUDYTE Report 2PR1WPO). Bayreuth, FRG: University of Bayreuth.
Putz-Osterloh, W., & Lemme, M. (1987). Knowledge and its intelligent application to problem solving. *The German Journal of Psychology, 11*, 286–303.
Reber, A.S. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General, 118*, 219–235.

Richardson, J. (1960). Comparison of S-R and R-S learning of paired associates. *Psychological Reports*, *7*, 225–228.

Roberts, F.S. (1976). *Discrete mathematical models. With applications to social, biological, and environmental problems*. Englewood Cliffs, NJ: Prentice-Hall.

Salomaa, A. (1985). *Computation and automata*. Cambridge: Cambridge University Press.

Sanderson, P.M. (1989). Verbalizable knowledge and skilled task performance: Association, dissociation, and mental models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 729–747.

Schumacher, R.M., & Gentner, D. (1988). Transfer of training as analogical mapping [Special issue: Human-computer interaction and cognitive engineering]. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-18*, 592–600.

Suppes, P. (1969). Stimulus-response theory of finite automata. *Journal of Mathematical Psychology*, *6*, 327–355.

Wagenaar, W.A., Keren, G., & Lichtenstein, S. (1988). Islanders and hostages: Deep and surface structures of decision problems. *Acta Psychologica*, *67*, 175–189.

Weir, G.R.S. (1991). Living with complex interactive systems. In G.R.S. Weir & J.L. Alty (Eds.), *Human computer interaction and complex systems* (pp. 1–21). New York: Academic Press.

Revised manuscript received 6 September 1992

APPENDIX

TABLE A1
Transition Matrix for the Small Automaton ("Ticket Vendor")

			Input Signal								
			alpha				beta				...†
			+				+				–†
			●	●●	●●●	●●●●	●	●●	●●●	●●●●	...†
State	Name	Output Signal	I1	I2	I3	I4	I5	I6	I7	I8	I9
S1	Wrong input	alpha, +, ○	1	2	1	3	1	1	1	1	9
S2	Card B	alpha, +, B	1	2	1	3	4	5	6	1	9
S3	Card D	alpha, +, D	1	2	1	3	4	5	6	1	9
S4	1 ECU	beta, +, 1	4	4	4	4	5	6	7	4	9
S5	2 ECU	beta, +, 2	5	5	5	5	6	7	7	5	9
S6	3 ECU	beta, +, 3	6	6	6	6	7	7	7	8	9
S7	Too much	beta, +, ~~~	7	7	7	7	7	7	7	7	9
S8	Eject card	**Goal**	1	1	1	1	1	1	1	1	9
S9	OFF	alpha, –, ○	1	2	1	3	1	1	1	1	9

†The "..." symbol indicates that any input is possible in this row.

Note: Only the output signals were presented to subjects, not the names of the states, which are included here for better readability. The horizontal sequence of the output signal elements corresponds to their vertical sequence in the actual display (cf. Figure 3). The labels of the three rows of input buttons are given in the three top rows on the right side.

TABLE A2
State Transition Matrix for the Complex Automaton ("Automatic Teller")

			Input Signal										
			•										○
			Amount					ΔΔΔ					...†
			A	B	C	D	E	A	B	C	D	E	...†
State	Name	Output Signal	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11
S1	NO CODE 0 ECU	•, Amount, 0	2	3	5	10	1	13	13	13	13	13	46
S2	10 ECU	•, Amount, 10	3	4	7	35	1	13	13	13	13	13	46
S3	20 ECU	•, Amount, 20	4	6	9	12	1	13	13	13	13	13	46
S4	30 ECU	•, Amount, 30	6	8	11	12	1	13	13	13	13	13	46
S5	35 ECU	•, Amount, 35	7	9	35	12	1	13	13	13	13	13	46
S6	40 ECU	•, Amount, 40	8	10	12	12	1	13	13	13	13	13	46
S7	45 ECU	•, Amount, 45	9	11	12	12	1	13	13	13	13	13	46
S8	50 ECU	•, Amount, 50	10	35	12	12	1	13	13	13	13	13	46
S9	55 ECU	•, Amount, 55	11	12	12	12	1	13	13	13	13	13	46
S10	60 ECU	•, Amount, 60	35	12	12	12	1	13	13	13	13	13	46
S11	65 ECU	•, Amount, 65	12	12	12	12	1	13	13	13	13	13	46
S12	TOO MUCH	•, Amount, ***	12	12	12	12	1	13	13	13	13	13	46
S13	NO Money, no Code	•, ΔΔΔ, ###	1	1	1	1	1	13	14	15	16	13	46
S14	B	•, ΔΔΔ, B___	1	1	1	1	1	13	13	17	13	14	46
S15	C	•, ΔΔΔ, C___	1	1	1	1	1	13	13	18	13	15	46
S16	D	•, ΔΔΔ, D___	1	1	1	1	1	13	19	13	13	16	46
S17	BC	•, ΔΔΔ, _C_	1	1	1	1	1	13	13	13	20	17	46
S18	CC	•, ΔΔΔ, _C_	1	1	1	1	1	13	13	13	21	18	46
S19	DB	•, ΔΔΔ, _B_	1	1	1	1	1	13	22	13	13	19	46
S20	BCD	•, ΔΔΔ, ___D	1	1	1	1	1	13	13	13	13	23	46
S21	CCD	•, ΔΔΔ, ___D	1	1	1	1	1	13	13	13	13	23	46
S22	DBB	•, ΔΔΔ, ___B	1	1	1	1	1	13	13	13	13	23	46
S23	Code OK: 0 ECU	•, OK;Amount, 0	24	25	27	32	23	23	23	23	23	23	46
S24	Code OK: 10 ECU	•, OK;Amount, 10	25	26	29	**45**	23	23	23	23	23	23	46
S25	Code OK: 20 ECU	•, OK;Amount, 20	26	28	31	34	23	23	23	23	23	23	46
S26	Code OK: 30 ECU	•, OK;Amount, 30	28	30	33	34	23	23	23	23	23	23	46
S27	Code OK: 35 ECU	•, OK;Amount, 35	29	31	**45**	34	23	23	23	23	23	23	46
S28	Code OK: 40 ECU	•, OK;Amount, 40	30	32	34	34	23	23	23	23	23	23	46
S29	Code OK: 45 ECU	•, OK;Amount, 45	31	33	34	34	23	23	23	23	23	23	46
S30	Code OK: 50 ECU	•, OK;Amount, 50	32	**45**	34	34	23	23	23	23	23	23	46
S31	Code OK: 55 ECU	•, OK;Amount, 55	33	34	34	34	23	23	23	23	23	23	46
S32	Code OK: 60 ECU	•, OK;Amount, 60	**45**	34	34	34	23	23	23	23	23	23	46
S33	Code OK: 65 ECU	•, OK;Amount, 65	34	34	34	34	23	23	23	23	23	23	46
S34	Code OK: too much	•, OK;Amount, ***	34	34	34	34	23	23	23	23	23	23	46
S35	Money OK, no Code	•, OK; ΔΔΔ, ###	35	35	35	35	35	35	36	37	38	35	46
S36	Money OK: B	•, OK; ΔΔΔ, B___	35	35	35	35	35	35	35	39	35	36	46
S37	Money OK: C	•, OK; ΔΔΔ, C___	35	35	35	35	35	35	35	40	35	37	46
S38	Money OK: D	•, OK; ΔΔΔ, D___	35	35	35	35	35	35	41	35	35	38	46
S39	Money OK: BC	•, OK; ΔΔΔ, _C_	35	35	35	35	35	35	35	35	42	39	46
S40	Money OK: CC	•, OK; ΔΔΔ, _C_	35	35	35	35	35	35	35	35	43	40	46
S41	Money OK: DB	•, OK; ΔΔΔ, _B_	35	35	35	35	35	35	44	35	35	41	46
S42	Money OK: BCD	•, OK; ΔΔΔ, ___D	35	35	35	35	35	35	35	35	35	**45**	46
S43	Money OK: CCD	•, OK; ΔΔΔ, ___D	35	35	35	35	35	35	35	35	35	**45**	46
S44	Money OK: DBB	•, OK; ΔΔΔ, ___B	35	35	35	35	35	35	35	35	35	**45**	46
S45	Money OK, code OK	Goal	1	1	1	1	1	1	1	1	1	1	46
S46	OFF	○, Amount, 0	1	1	1	1	1	1	1	1	1	1	46

†The "..." symbol indicates that any input is possible in this row.
Note: Description analogous to Table A1.

TABLE A3
Recognition Items Used in the Diagnostic Phases

Item	Small Automaton S_t	I_t	S_{t+1}	Comment	Complex Automaton S_t	I_t	S_{t+1}
1	1	4	3	Code → Code	37	8	40
2	2	2	2	Code → Code	38	7	41
3	2	6	5	Code → ECU (mode interaction)	14	1	1
4	3	7	6	Code → ECU (mode interaction)	20	10	23
5	1	9	9	Code → OFF (on/off interaction)	38	11	46
6	4	9	9	ECU → OFF (on/off interaction)	10	11	46
7	4	1	4	ECU → ECU (recursion)	23	5	23
8	7	3	7	ECU → ECU (recursion)	34	2	34
9	4	5	5	ECU → ECU (less efficient)	6	1	8
10	4	6	6	ECU → ECU (more efficient)	1	3	5

Note: The numbers of the states and input signals refer to the state transition matrices depicted in Tables A1 and A2. The "Comment" column illustrates which functional aspect of the automaton is reflected in the item.