GENERATING SCRIPTS FROM MEMORY[1]

by

Walter Kintsch & Suzanne M. Mannes

University of Colorado

Abstract

A variation of the Raaijmaker and Shiffrin (1981) retrieval model is proposed to account for typical script generation data. In our model, knowledge is represented as an associative network with propositions as the nodes. A control process which utilizes temporal information in these propositions supplements the probabilistic memory retrieval process to produce ordered retrieval of scriptal events. Simulations are reported which provide a good qualitative fit to data collected from subjects in both script generation and free association tasks. These results support a view of memory as an unorganized knowledge base rather than a stable, organized structure.

1. Scripts as Mental Structures

Scripts are representations of simple stereotyped event sequences. As such, they are a subtype of frames or schemata. In order to be useful, knowledge has to be organized in some way, and scripts, as well as frames, schemata, and semantic nets, provided that organization (e.g. Anderson, 1980; Graesser, 1981; Schank, 1980; Schank & Abelson, 1977). Scripts are claimed to be mental structures, and psychologists set out to demonstrate the psychological reality of these structures and to investigate their properties. Bower, Black, & Turner (1979) observed a high level of agreement when subjects were asked to list the characteristic events that they thought belonged to a script, and their order.

However, the view that scripts are precompiled mental structures was soon challenged from several sides. Computationally, fixed mental structures like scripts turned out to be too inflexible to really serve the purposes for which they were originally designed (van Dijk & Kintsch, 1983; Schank, 1982). If scripts guide retrieval, how close events occur in the script structure should determine the time it takes to retrieve one event, given the other. However, such distance effects have not been observed (Haberlandt & Bingham, 1984; Galambos & Rips, 1982; Bower et al., 1979). At most, it appears that immediately succeeding events are retrieved faster than events farther away (Bower et al.), but that argues more for a local relation like "next", rather than for a larger script structure.

For these reasons, there has been a general move away from considering scripts as fixed mental structures. In the present paper we claim that knowledge is not pre-organized in terms of scripts and schemata, but that such structures are generated from an unorganized associative net in response to a specific task demand in a specific context. Only in this way can the flexibility

[1] This research was supported by grant number MH - 15872 from the National Institute of Mental Health. It is publication number 87-3 of the Institue of Cognitive Science at the University of Colorado, Boulder, CO 80309

and context sensitivity that characterize human script use be achieved. Here, we shall show how a behavior that has been taken as prima-facie evidence for the existence of scripts as mental structures can be generated from a knowledge base in which there are no pre-existing global structures like scripts. Specifically, we shall simulate how people go about listing the events which constitute common scriptal activities, such as going to a grocery store.

2. Scripts and Categories

The approach taken here extends earlier work on generating conceptual categories to scripts. Walker & Kintsch (1985) have modelled this process by making two crucial assumptions.

First of all, they assumed that knowledge retrieval obeys the same laws as retrieval from episodic memory. In particular, Walker & Kintsch assumed that the automatic component of the retrieval process can be described by the Raaijmakers & Shiffrin (1981) theory. Given a particular retrieval cue, this model predicts what will be retrieved, and when. On the other hand, the control processes which are necessary to put together an appropriate retrieval cue in the present situation are outside the Raaijmakers & Shiffrin model, and will be discussed below.

Secondly, Walker & Kintsch assumed that the retrieval process operates on an associative net in which categories are not explicitly represented. Of course, there must be information stored with each category member that identifies it as such. In the simplest case this might be an associated IS-A proposition.

A model of script generation will be outlined below which is analogous to the category generation model of Walker & Kintsch, except that it utilizes a different control process to reconstitute exhausted retrieval cues: instead of randomly picking an associated node and adding it to the retrieval cue, local information about what comes next is used. In this way the unproductive search phases which are characteristic of category retrieval are avoided, and an essentially linear retrieval function is obtained. Furthermore, the retrieval process does not slowly peter out, but stops when the end of the chain is reached. Before describing this model in more detail, however, it is necessary to look more closely at how people generate scripts, so that we have a more solid data base with which to compare our model.

3. Generating Scripts: Experimental Results

3.1 Method

Six subjects were asked to think aloud (Ericsson & Simon, 1984) as they pretended to tell a stranger from another culture what typically happens in the following three situations: *going to a restaurant for a meal, going to a grocery store to buy groceries, and going to a doctor's office for a checkup*. These were the same scripts that were used by Walker & Kintsch (1985).

3.2 Results

Each subject's protocol was divided into idea units, roughly corresponding to propositions in the sense of van Dijk & Kintsch (1983).These units were classified as either scriptal events or elaborations. Events were always single propositions, while elaborations were sometimes more complex. For example, the event *make a list* was elaborated with *of what you want to buy*.

There was good agreement among subjects about the events they thought belonged to each of the three scripts. Figure 1 shows the percentage of all responses which were produced by 5 or 6 subjects, 3 or 4 subjects, or by only 1 or 2 subjects. More than half of all responses were given by a majority of the subjects. Almost all items were generated in their natural order; only 1% of all responses were out of order. Figure 2 shows the cumulative number of scriptal units as a function of retrieval time for one of our subjects. The important points to notice are first that the rate at which a script is generated is approximately constant. Secondly, the curve is relatively smooth, in contrast to the severe scalloping observed for category naming tasks.

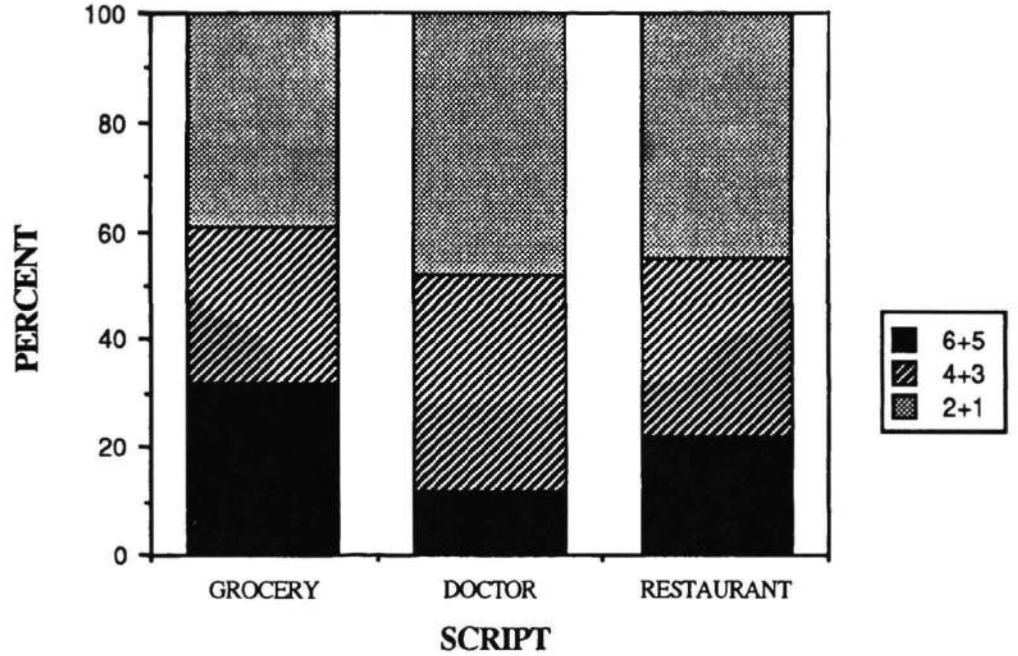

Figure 1. Percent of events agreed upon by 6+5, 4+3, and 2+1 subjects.

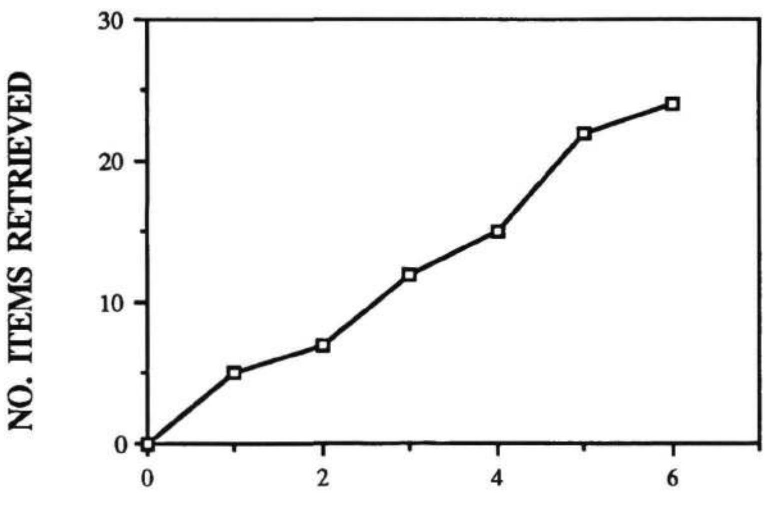

Figure 2. Number of grocery store items retrieved by a single subject as a function of time (plotted in 30 second intervals).

In listening to these protocols, one is struck by their fluency. Subjects always seem to know what to say next, with little hesitation. There was apparently no need for an extended search for a new retrieval cue. We therefore closely inspected each protocol for possible evidence as to the nature of the retrieval cues that permitted such smooth transition from event to event. In the course of these analyses we arrived at the distinction between scriptal events and elaborations. As already mentioned, events are single-proposition units, expressing an action either by the main actor or by some other participant (checkout-clerk, nurse, waiter) which can stand on their own. That is, events were meaningful by themselves, and did not require reference to some other unit for their interpretation. Elaborations on the other hand, have to be interpreted in the context of some other unit, usually an event. The event *make a list* was elaborated by *of what you want to buy*. Thus, elaborations appear to be dependent on events. Events, however, either follow directly previous events, or are preceded by explicit temporal connectives, such as *then*, or *after that*. A striking asymmetry was noted in this respect: temporal connectives were used by most or all of the subjects in certain places, while they were rare otherwise. For instance, each subject used a temporal marker (connective) between the last item that dealt with preparation for shopping and entering the store and the first item that dealt with the actual shopping. The transition between shopping and checking-out was similarly marked by all subjects, and 5 of the 6 subjects separated checking-out and leaving the store with a temporal marker. Within each of these episodes, in contrast, temporal connectives were rare and used idiosyncratically. Thus, explicit temporal connectives appear to segment the script into separate episodes which are crucial for an understanding of how scripts can be generated from an associative knowledge base. The distribution of these temporal markers is shown in Figure 3.

3.3 A Hypothesis about Retrieval

In Table 1 the events and elaborations for the grocery store script which were produced by our subjects are shown, broken down into episodes as suggested by Figure 3. Items generated by 5 or 6 subjects are shown with asterisks; inferred items (the episode labels) are shown in brackets. The reponses made by one subject are connected by a continuous line, to show this subject's retrieval path.

We are now in a position to state a possible hypothesis about the retrieval cues which control the process of script generation. The answer is simple for elaborations: since all elaborations can be co-ordinated with a specific event, we assume that the events serve as their retrieval cue. For the events themselves, we propose a dual process: some events are retrieved via specific temporal information, while others are retrieved associatively, much as category members are. Our data suggest that goal-directed retrieval on the basis of specific temporal cues, such as X FOLLOWS Y operates at the level of episodes: when enough information within an episode has been generated, the subject does not search for a new retrieval cue by checking random associations, but uses specific temporal information to establish a new episode cue. Thus, once the subject is done with *checking-out*, the knowledge base is searched for a proposition AFTER[CHECK-OUT, $], yielding AFTER[CHECK-OUT, LEAVING], and LEAVING becomes the next retrieval cue. The episode cue itself works much like the category cue in the category naming task: repeated retrieval attempts are made using this cue in conjunction with recently retrieved information. Thus, within-episode retrieval is locally governed, and stops when a certain number of unsuccessful retrieval events have occurred. At that point, a new episode cue is generated in the manner discussed above. Temporal information available in the knowledge base specifies what that cue has to be, and hence a long search is superfluous, giving the script generation process its characteristic smoothness and fluidity. In the next section, this hypothesis about how scripts are generated will be specified further to the point where simulations can be performed. The goal of these simulations will be to determine whether simulated script retrieval is at least qualitatively similar to the human data.

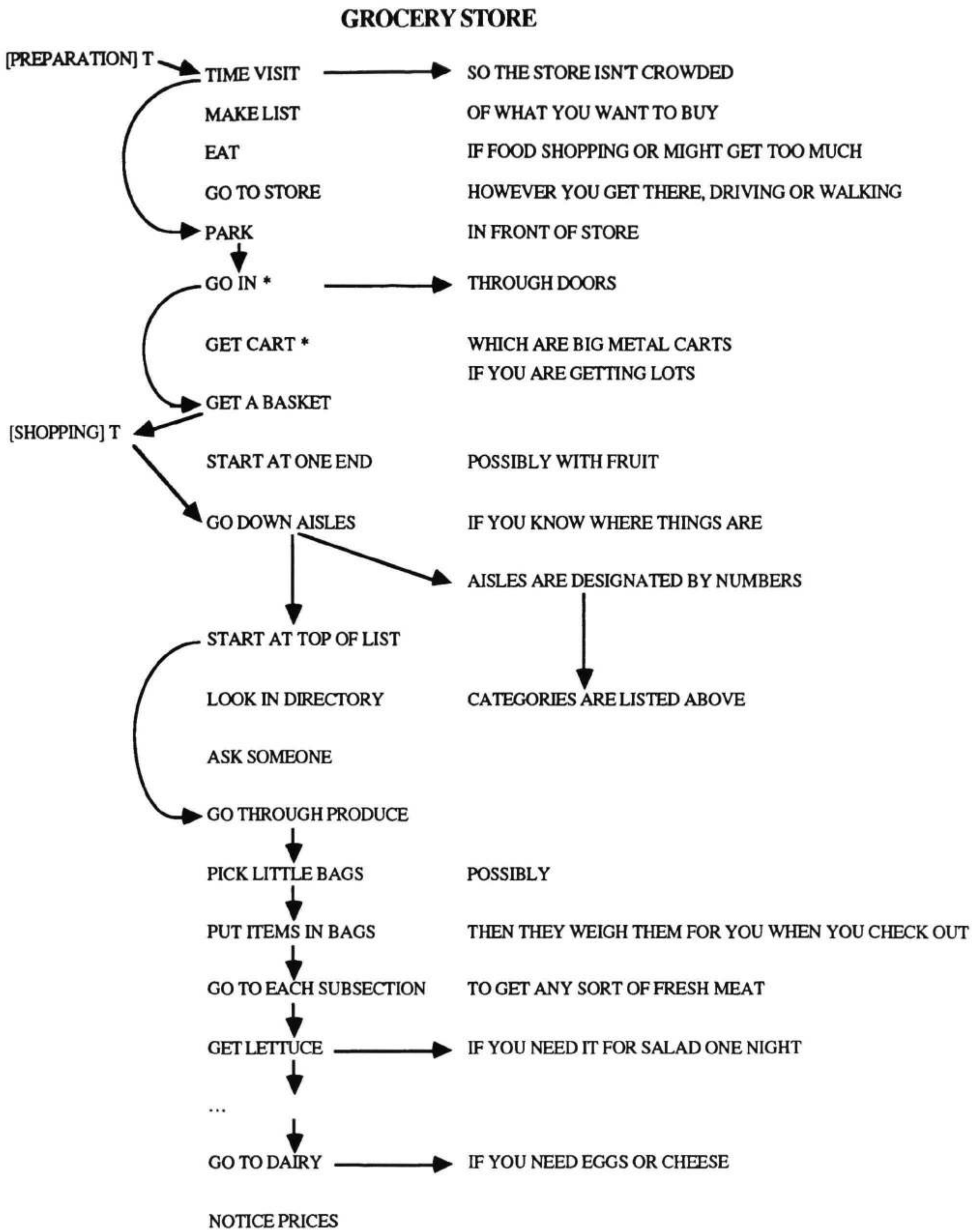

Table 1. Items which subjects produced in response to the Grocery Shopping cue. The lines show one subject's path through the network. See text for further explanation.

	GO TO FRONT	THERE ARE MAGAZINE RACKS
[CHECKING OUT] T *		
	GO TO CHECK OUT	THERE'S A CASHIER
	TAKE OUT GROCERIES	
	PUT BASKET UP	
	MACHINE READS CODES	
	SEE PRODUCTS OVER SCANNER	EVERY ITEM HAS A CODE
		PRICES ARE REGISTERED IN COMPUTER
	RING UP ITEMS	
	SEE ITEMS DOWN BELT	THERE'S A SACKER THERE
	CLERK PUTS GROCERIES IN BAGS *	DEPENDING ON THE NUMBER OF ITEMS
	CLERK PUTS BAGS IN CART	
[PAY/LEAVE] T *		
	CASHIER TELLS YOU THE AMOUNT *	
	YOU TAKE WALLET OUT	
	WRITE CHECK	
	PAY *	THE RIGHT AMOUNT YOU OWE
	GET CHANGE	
	WALK OUT	
	GET HELP WITH BAGS	IF YOU'RE OLD OR JUST LAZY
	CARRY BAGS OUT	
	LOAD CAR	
	GO HOME *	

Table 1. (continued).

4. Generating Scripts: A Simulation

Memory is an associative network with concepts and propositions as nodes. Nodes are related either positively, negatively, or not at all, with connections ranging from -1 to +1. Mathematically, this network will be represented by a matrix, **K**. The rows *(i)* and columns *(j)* of this matrix represent the nodes of the network, and the entries $(s_{i,j})$ represent the connection strengths between the nodes.

A retrieval cue consists of one or more nodes of this network. Each retrieval attempt with a particular retrieval cue results in the retrieval of a single node from the network (though not necessarily in an overt response). The model can be concisely statedin terms of the well-known model of memory retrieval proposed by Raaijmakers & Shiffrin (1981). Memory in their model, is represented as an associative net, and the retrieval process which they describe results in the following equation.

$$(1) \qquad Pr\,(j \,/\, i_{1...k}) = \frac{\prod_{i=1}^{k} s_{i,j}}{\sum_{j=1}^{n} \prod_{i=1}^{k} s_{i,j}}$$

Suppose that node *j* has now been retrieved. If it has not already been retrieved previously, it will be output as an overt response. An unsuccessful retrieval attempt occurs when the retrieved node duplicates an earlier retrieval.[2]

The newly retrieved node is added to the retrieval cue, which is now of size $k+1$. If this number exceeds some maximum value *m*, an old element must be dropped. We assume that this will be the most recently added element. Thus, the nature of the retrieval cue changes dynamically in response to local effects, but at the same time retains a stable core.

Free association differs from script generation only in that no temporal cues are used to guide retrieval. *H* is used to retrieve some associate which then takes on the role H_1. After it has become ineffective, some other associate of *H* takes its place. Thus, we obtain clusters of related associates, with occasional unpredictable jumps to new ones.

4.1 Simulating Grocery Shopping

In order to test the model outlined above, we attempted to simulate the generation of a grocery-shopping script. The goal of this simulation was to account qualitatively for the data described in Section 3. Specifically, we wanted to see whether the simulation yielded an output comparable to the human data in Table 1, and whether the rate with which it was produced was constant, as in Figure 2. For this purpose, an associative knowledge system containing information about grocery shopping had to be constructed first.

4.1.1 The Knowledge Base.

Since merely a qualitative account is attempted here, we need not concern ourselves with all the knowledge people have about grocery shopping. Instead, fifteen typical items from Table 1 were selected more or less randomly and grouped by episodes. In addition, the script heading, the four episode headings as well as the temporal markers between them were included in the

[2] In other cases this editing process must be more complex. E.g., in category naming a check needs to be made to determine whether the retrieved node is, in fact, a category member.

knowledge base. Furthermore, for each of the 15 scriptal events plus the 5 headings free associations were obtained from a group of 16 subjects. The two most frequently produced associations for each item were also added to the knowledge base. All the items in the knowledge base thus far are related to *grocery shopping*. To make the simulation more challenging, items unrelated to the grocery script are needed, to see whether or not the retrieval process suffers interference from such items. Therefore, a closely related script was selected, for which three of the four episodes overlapped. Subjects were asked to provide us with a *shoe-store* script. This yielded the three episodes *Entering, Shopping,* and *Leaving*. In addition to the three episode headings, four of the produced scriptal events are associated with both scripts. If the two scripts will interfere with each other, there is certainly opportunity to do so. For each of the *shoe-store* events, two high-frequency associates were also added to the knowledge base. There is, of course, in the knowledge base no distinction between scriptal events, episode headers, or associates: all are just nodes in a network, - we categorize them in this way only on the basis of the protocols subjects generate.

The total matrix thus constructed had 63 rows and columns - a tiny fragment of a human knowledge system. The 3,969 connection strengths $s_{i,j}$ in this matrix were estimated from actual script and free association data. Only rough estimates were made: Whenever a stimulus elicited the same response in 75% of the subjects, a strength value of 1 was used; responses given by fewer subjects were assigned a connection strength of .5; and responses which did not occur in our sample were given a strength of 0. All associations were assumed to be symmetric.

In addition, connections were established on the basis of the script data in the following way. The script headers *grocery shopping* and *shoe store* were connected with a strength of 1 to their respective episode headers and with a value of .5 to the scriptal events within each episode. The episode headers were connected with a strength of 1 to their respective script events. Finally, script events within each episode were connected to each other by a value of .5. However, the strengths of these connections were not necessarily symmetric: *picking up vegetables* and *going through the aisles* are connected both ways, and both are connected to *going to the check-out counter*, but the latter has no strength connecting it to either of the two former nodes. Thus, $s_{(vegetables,aisles)} = s_{(aisles,vegetables)}$, but $s_{(aisles,checkout)} \neq s_{(checkout, aisles)}$. Furthermore, the backwards connections between episode headers and their script headers, as well as between script events and their episode headers, were made less strong (.5) than the forward connections between these elements. Thus, scriptal events may be directional. (It is of course possible that the same may be true for associations in general, but this possibility was not explored here.) The temporal connectives - BEGIN[GROCERY-SHOPPING, ENTER], BEGIN[SHOE-SHOPPING, ENTER], AFTER[ENTER, SHOPPING], AFTER[SHOPPING, CHECK-OUT], AFTER[SHOPPING, LEAVE], AFTER[CHECK-OUT, LEAVE] - which are needed to control the retrieval process in the script generation task are connected by a value of +1 to their second argument, and a value of -1 to their first argument. The decision to use only the values 1 and .5 for connection strengths is both crude and arbitrary, but by restricting ourselves to such rough approximations we have reduced the need for subjective judgments. It should also be noted that precise strength values matter relatively little here: It is the over-all pattern of connections that determines the results.

4.1.2 **Retrieval**.

Instead of estimating the parameters of the model from the data and trying to fit the data quantitatively, all the parameters were specified a priori.[3] Specifically, we assume that the retrieval cue has three components, two stable (which would be the script header and an episode

[3] While it would be possible to estimate the parameters of the retrieval model statistically, a quantitative fit would presume an adequate simulation of the knowledge base, which is beyond our possibilities at present.

header) and the third one variable (the last item retrieved). We also assume L = 3, that is, a retrieval cue is abandoned after three successive retrieval failures. Consider a particular simulation run with this model, then. What we observe is an interplay between the controlled search for a new retrieval cue, and the automatic retrieval process. Given Grocery Shopping, the proposition BEGIN[GROCERY-SHOPPING, $] is used to retrieve ENTER via a pattern matching process. These two nodes then form the first retrieval cue, activating 7 nodes in the system to some extent or other. *Entering Store* is selected. It is produced as a response, and becomes the third component of the retrieval cue. This modified cue now activates four other nodes, and, after 2 failures (the cue retrieves a component of itself), *Getting a Cart* is retrieved. This now replaces *Entering Store* as a component of the retrieval cue, which retrieves, after one failed attempt, a node which entered the network as an elaboration of Getting a Cart: *is fun*. The new retrieval cue activates 5 nodes, 3 of them new ones, but fails anyhow, because the same, old node is retrieved coincidentally three times in a row. Now the control process takes over, again: A memory search is made on the basis of AFTER[ENTER, $], which yields SHOPPING as the new episode cue, and the process once again shifts into its automatic retrieval mode. This interplay continues until the last episode cue is exhausted. In the simulation run under discussion here, a total of 12 nodes is thus retrieved, among them 5 nodes which we have called - on the basis of the data our subjects had given us - scriptal events; the rest were episode headers and elaborations. The data from this simulation run are shown in Table 2. This network looks much like an abbreviated version of Table 1: obviously, what the model does is quite close to what people do when they generate scripts. Most importantly, the retrieval process stayed on track: no intrusions from the shoe-store domain occurred, in spite of the fact that several of the items produced as part of the grocery script were also associated with the former domain. Secondly, the items were produced in the correct order, wherever order mattered, as in the actual data.

Figure 4 shows the results of three independent simulation runs with the grocery script. These simulation results have the same features as the data obtained from the individual subject in the script generation task (Figure 2): the rate at which scriptal events are produced is constant, there is no slowing down towards the end, and the curves are relatively smooth, without the large scallops which characterize category retrieval.

4.1.3 **Free Associations.**

Figure 5 shows the rate at which free associations to both grocery store and shoe store are produced by the model. To arrive at this figure, a different control process was assumed than for script generation, which does not involves the use of temporal cues. Given the script header, an associated item was produced. These two items then formed the first retrieval cue. As the third item was retrieved it was added to this retrieval cue, but the next item retrieved replaced it, and so on. After three unsuccessful retrieval attempts, this cue was abandoned, and the process was started all over again with the script header as the sole component of the retrieval cue. Once the script header itself lead to three successive retrieval failures, the whole process stopped.

The retrieval functions shown in Figure 5 are negatively accelerated and somewhat scalloped. They look more like a category naming function than a script generation function. Nevertheless, they were generated from the same knowledge base, by the same retrieval process as Figure 4 - only the control process was different. Scripts, categories, and associative structures do not reflect the organization of memory; rather, they are generated from an unorganized knowledge base - an associative net with only local connections between concepts and propositions. Structures need not be in the mind; they may result from particular kind of control processes - that is what this model suggests.

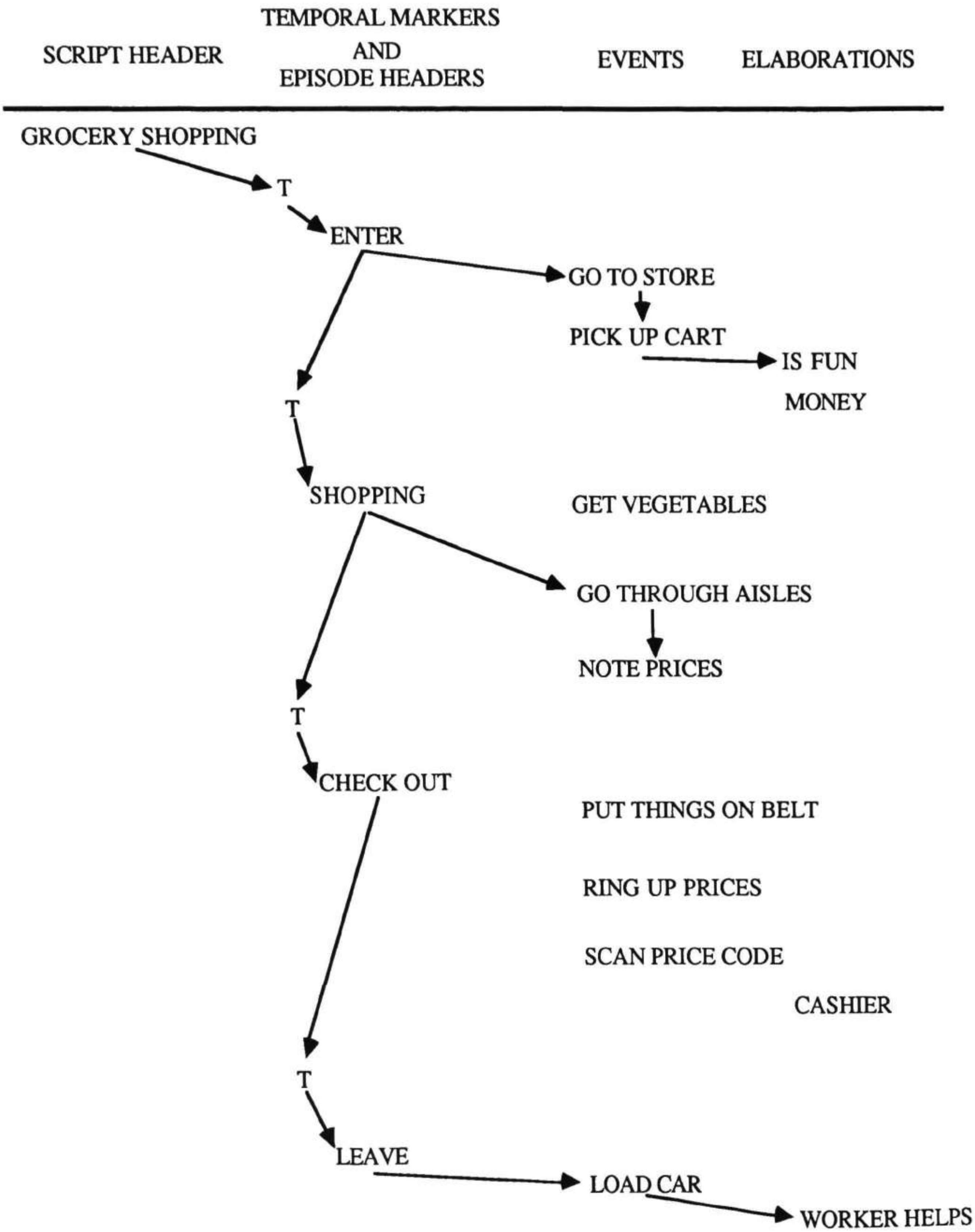

Table 2. Items which the model retrieved over three runs in response to the Grocery Shopping cue. The lines show the result of one run.

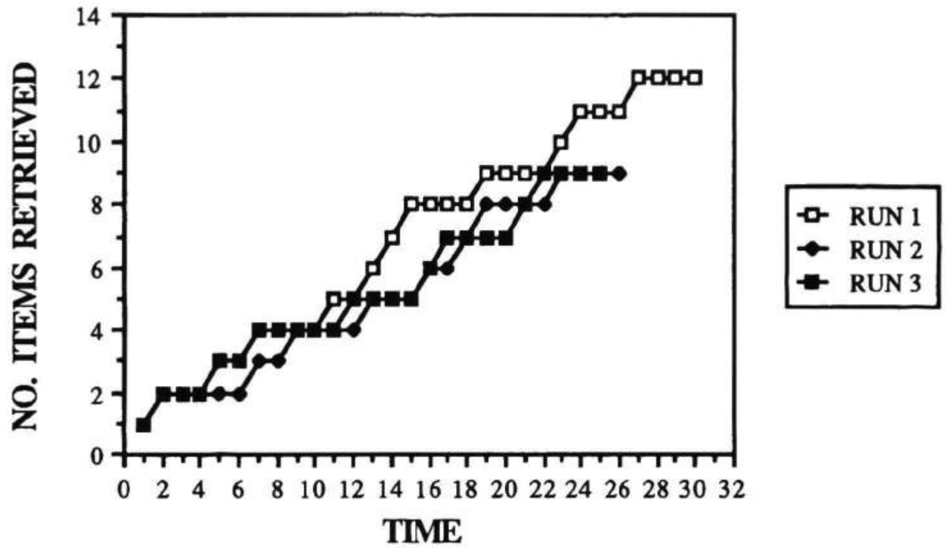

Figure 4. Number of scriptal items retrieved by the model in three runs as a function of time.

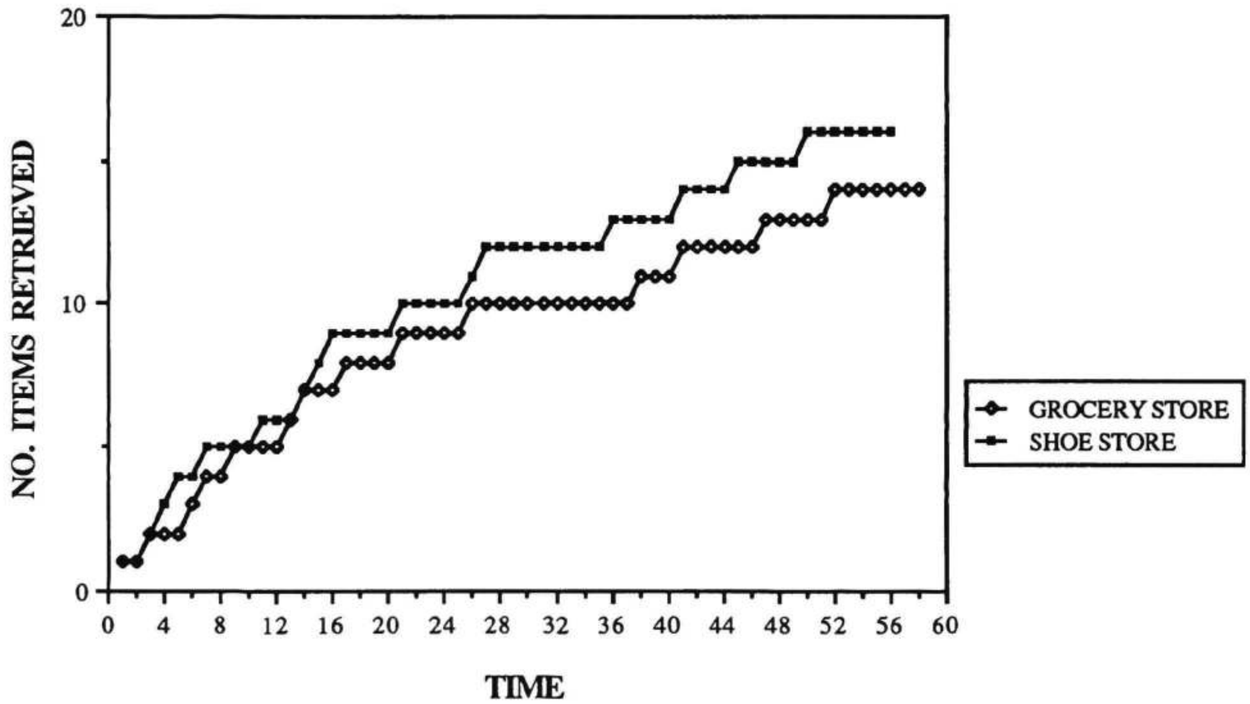

Figure 5. Number of items retrieved by the model in a free association task as a function of time.

References

Anderson, J. R. (1980). Concepts, propositions, and schemata: What are the cognitive units? *Nebraska Symposium on Motivation, 28,* 121-162.

Bower, G. H., Black, J B., & Turner, T. J. (1979). Scripts in memory for text. *Cognitive Psychology, 11*, 177-220.

Ericsson, K. A. & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data.* Cambridge, MA: MIT Press.

Galambos, J. A. & Rips, L. J. (1982). Memory for routines. *Journal of Verbal Learning and Verbal Behavior, 21*, 260-281.

Graesser, A. C. (1981). *Prose comprehension beyond the word.* New York: Springer-Verlag.

Haberlandt, K. & Bingham, G. (1984). The effect of input direction on the processing of script statements. *Journal of Verbal Learning and Verbal Behavior, 23*, 162-177.

Raaijmakers, J. G. & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review, 88*, 93-134.

Schank, R. C. (1980). Language and memory. *Cognitive Science, 4*, 243-284.

Schank, R. C. (1982). *Dynamic memory.* New York: Cambridge University Press.

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

van Dijk, T. A. & Kintsch, W. (1983). *Strategies of discourse comprehension.* New York: Academic Press.

Walker, W. H. & Kintsch, W. (1985). Automatic and strategic aspects of knowledge retrieval. *Cognitive Science, 9*, 261-283.