19

Designing a Learner: Some Questions

David Klahr

Carnegie-Mellon University

What do we know about what happens to a learner during instruction? One of the most exacting criteria for testing our knowledge about any phenomenon is the extent to which we can build a model that exhibits the behavior being studied. If we can simulate it, then we have at least a sufficiency model. Of course, there may be many aspects of the model that lack plausibility, but they can then become the focus of further study. (Reitman, 1967, once characterized this simulation approach to cognitive psychology as a way to "invent what you need to know.") In this chapter I will raise some questions about how one might go about building a model of a learner in an instructional mode (MOLIM).

LEARNING AS PROBLEM SOLVING

To learn is to solve a problem. In all but the most elementary situations, learning is under the learner's strategic control of attention and memory. If this view of learning is valid, then the study of complex problem solving—and the orientation such study provides to cognitive psychology—has direct relevance for the design of a MOLIM. In this section I will mention a few features of problem-solving theory that seem to justify the view of learning as problem solving, and that also have particular importance for the study of learning. Then, in the next section I will raise some questions about the design of a MOLIM.

Current information processing approaches to the study of human problem solving proceed by postulating a general system architecture, and then constructing explicit representations for the data structures and the processes that generate the observed problem solving behavior. A problem solution consists of an internal representation for some knowledge that the system did not have at the outset. Problem solving consists of a series of local transformations of knowledge that ultimately reach the desired knowledge state. In several of the chapters in this volume (e.g., the Greeno's Chapter 7), the "solution" to the learning problem is explicitly represented as a data structure (semantic net) and a set of procedures for searching that network. But note that these results—these solutions to the learning problem—are static with respect to the learning process itself. That is, with respect to the time grain of the instructional process, the results of instruction, even though they may themselves by dynamic processes, are structures upon with the learning system must operate. We need a model of the system's response to instruction, that is, its functioning in circumstances in which it must attend to the instructional episode and modify its own performance structures and processes.

In our instructional efforts, we try to provide optimal environments for the human information processing system to learn something. As it is with the horse led to water, so it is with the learner in an instructional situation: we can't make it ingest what we offer. The instructional design question is typically "*Will* the learner learn from this instruction?" A further question should be "*Why* should he learn?" The view of learning as problem solving suggests some ways to characterize this question. Problem-solving theory (Newell & Simon, 1972) includes two features of importance for our purpose. One is a detailed internal representation of the task environment. The other is a characterization of how the human information processor allocates its limited processing capacity to the problem-solving process. A principal method for effecting this allocation is the use of explicit representations for goals. Goals are symbolic expressions that direct and control the course of problem solving, representing what the system "wants" to do at any moment, and "why" it wants to do it. Thus, the answer to whether or why the system will learn becomes, in view of learning as problem solving, a matter of stating the circumstances under which learning-related goals are generated and manipulated.

SOME DESIGN QUESTIONS FOR A MOLIM

In this section, I will raise four questions that must be answered by the designer of a MOLIM:

1. When should learning occur?
2. How will the system be changed as a result of learning?
3. How thoroughly assimilated is the thing to be learned?
4. How distinct is learning from performance?

These questions, and their answers, are highly interrelated, and it is difficult to determine their appropriate order of presentation. Their ordering here is arbitrary, and does not imply any particular differential importance in my mind.

1. When should learning occur? A curious problem with most of the learning models in both cognitive psychology and artificial intellignece is that they are too single-minded in their task: they learn all the time. In designing a plausible MOLIM, we must be able to account for the fact that most of the time learning does *not* occur. We can do this by explicitly including in our MOLIM the precise conditions under which an instructional episode causes something to be learned. This is where the appropriate use of goals could play a role. Rather than construct a system in which the tendency to learn is integrally built in to the underlying operating mechanisms, we can design a more general problem solver whose problem is to learn, and whose goals include explicit learning efforts.

By themselves, such goals would still be inadequate for deciding when the system should learn. Additional information would be required about the current state of knowledge—that is, about both the current configuration of the external environment as well as the internal state of the system Thus, another design decision concerns those variables and their critical ranges which would, in conjunction with the learning goals, activate the self-modification processes.

The mechanisms that determine when learning is to occur must be capable of representing differential responsiveness to instruction. As Resnick (Chapter 3, this volume) has pointed out, our models must be able to represent both early and late forms of task proficiency, and for a MOLIM, the task is learning itself. Therefore a MOLIM must incorporate the capacity to represent both early and late learning proficiencies. Siegler (1975) has noted the importance of experimental designs in which both older and younger children are given the same training sequences, in order to examine the possible interaction of age and instructional effects. Since such interactions have been found (e.g., Siegler & Liebert, 1975), we must be able to represent them in MOLIM, through the general strategy, suggested by Resnick, of building developmentally tractable models.

2. How will the system be changed as a result of learning? This is, perhaps, a more useful way to say "what is learned?" There are several outcomes that can result from the learning effort. One result is that nothing happens: the learning attempt fails, and no lasting change is made in the system As noted above, this is more the rule than the exception in real instructional situations, and we must be able to build a system that can handle this fact.

Another possible result is that the entire system architecture could change. That is, the system's components and their interrelationships might be altered. However, since by "system architecture" I mean "hardware" rather than the "software" of the human information processor, it seems unlikely that this kind of change is really the result of instruction. Although it would be required in full developmental theory, we need not be too concerned with it for now.

There are two kinds of software changes that the system can undergo: changes in processes and changes in structures. Newell (1972a) has demonstrated the imprecise nature of the process–structure distinction in systems that are themselves undergoing change. In the case of a self modifying system, the ambiguity becomes even greater, since it is linked to the issue of degree of assimilation to be described below. Although one often makes an apparently unambiguous distinction between instruction directed to acquisition of facts, and instruction directed to the teaching of procedures (or skills), it is clear from the work of Greeno, Hyman, and Norman (Chapters 7, 8, and 9 of this volume) that the issues are not so simple. One can represent factual knowledge by procedures that can generate those facts, and, conversely, one can represent what could be procedural outputs by appropriately complex static symbolic networks.

Another type of change that may result from instruction, and which we must therefore be prepared to explicitly represent in our MOLIM, is change in the learning properties of the system beyond the representation of the specific instructional material. For example, in the case of the aggregate models with which Atkinson represents the learner, there are a few changes in the acquisition parameters that result from the instruction. In more complex models of learning, such systemic modifications would include the basic rules of self-modification themselves.

3. How thoroughly assimilated is the thing to be learned? In the paper cited above, Newell (1972) distinguishes between several levels of general versus specific knowledge about a task. The more general the knowledge, the more transformational rules are necessary to take the system from its entry state at performance time to a task-specific state in which it can actually perform the task at hand. Conversly, a very task-specific piece of knowledge might be represented in "machine code": being fully assimilated it would require no interpretation at run time, however it would be of limited generality.

A concrete example of this distinction is provided by the models for children's performance on seriation tasks developed by Baylor and Gascon (1974). In these models there are two kinds of representations for "seriation knowledge." One is a base strategy, consisting only of series of nested goals, that describe, at the most general level a strategy for seriation (e.g., "find max," or "insertion"). The other representation is a rule set that accounts for each move made by the child during a specific seriation task. Behavior during length seriation has one rule set, and behavior during weight seriation has another. If the system has only the base strategy, then it also requires a set of rules that take the base strategy and construct a task-specific variant (e.g., for weight seriation). There are various ways to conceptualize this mapping. The two simplist are a complete "compilation," in which the base strategy, plus the task-specific mapping rules, create an entire task-specific system that then runs on the task. The other is a collection of interpretive rules that never create a task-specific entity, but instead interpret the base strategy, "on the run," in terms of the specific task.

In designing a MOLIM, we must decide upon the assimilatedness of the information to be acquired. The sematic networks of Norman and Greeno in this volume appear to be far toward the task-specific end of the spectrum, while Shaw and Wilson appear to be focusing upon a more general "base strategy" in their representations of group generators. A similar contrast can be found in comparing Atkinson, (Chapter 4) with Resnick (Chapter 3). Atkinson is aiming at an instructional procedure that will create a very specific set of data structures and processes that will enable the learner to write programs or to acquire a set of reading patterns. Resnick has begun to investigate the manner in which the learner abandons the task-specific instructions and creates a more efficient and general procedure. My strategic bet is that by representing the result of learning as "base plus interpreter," we may begin to get a handle on the mechanisms of generalization from, or beyond, the specific instructional sequence.

4. How distinct is the learning system from the performance system? In almost all models of learning, be they psychological models or examples of Artificial Intelligence, there is a clear distinction between the learning processes and the thing to be learned, that is, the performance system (see for example the models in Feigenbaum & Feldman, 1963, or Simon & Siklossy, 1972). The distinctions are made with respect to the over-all organization of the respective systems, the underlying representations, and even the basic system architecture. For example, in the letter series completion model of Simon and Kotovksy (1963), much attention is paid to the differential short-term memory demands made by different representations for different serial concepts, but the demands made during the induction of these concepts (i.e., during their learning) are not directly addressed. Another example of this distinction can be found in Waterman's (1970) learning program in which the result of training was represented as a production system, but the learning system itself was not a production system.

Although such separation has the benefit of making the modeling task more manageable, it lacks both elegance and psychological plausibility. I would hazard the guess that the same mechanisms that span the gap between general base strategy and the task-specific system (see Question 3, above) are implicated in the learning process itself. The more homogeneously we design the MOLIM, the more likely we are to be able to solve both problems simultaneously. Such a view might be nothing more than idle speculation were it not for the recent work of my colleague, Don Waterman. He has constructed a set of adaptive production systems for a range of learning tasks (Waterman, 1974a b). These models learn simple addition, verbal associations, and complex letter series. Each model is written as in initial core of productions, some of which have the capacity to add additional productions to the initial core. The final "learned" system operates under the same control structure and system architecture as the initial system and the learning rules are represented in precisely the same way as the new rules that are learned, that is, as productions.

The instructional environments in which Waterman's system do their self-modification are relatively simple, but I believe that the basic approach is very sound, and extendable to richer instructional problems. In a less precise but much more general statement, Wallace and I (Klahr & Wallace, 1976) have proposed a broad view of cognitive development in terms of a self-modifying production system.

CONCLUSION

The chapters in this volume represent diverse but converging answers to the question of the relevance of some current research in cognitive psychology to instructional design. I have not attempted to synthesize, evaluate, or review the previous chapters because several efforts have already been made in the discussion chapters by Gregg, Olson, Farnham-Diggory, Hayes, Collins, Shaw, Glaser, and Cazden. Instead, my intent has been to provide an orientation that might help the reader to form his own evaluation of the research reported here.

By confining attention to learning in intentionally instructional situations, I have attempted to reduce the task to manageable proportions. Instead of concocting a general learning system, I have considered the design of more limited models of the effects of specific instructional situations. Such models will initially tend to be largely determined by the task environment, that is, by the material and its form of presentation. However, there are some fundamental questions that are worth asking, questions which may apply to a wide range of instruction, even though their answers may be task-specific.

Having posed the design issues, we might ask a few questions about the enterprise, per se. Why bother with such an effort? There seem to be a few good reasons. First, if we could actually build a sensible model, we could directly simulate the results of proposed instructional procedures. The potential value of such instructional "pilot plants" is that they could replace the extensive field testing of instructional variations that we are presently forced to use. But except in the most simple situations, we are not yet able to build such models, and the worth of the enterprise lies in its propaedeutic nature: it may give us an introduction to the kinds of things we still need to know. Several of the comments by other discussants in this volume have raised the disquieting possibility that we may have little here that is really new or useful. I think that such a view is unjustified, but the issue cannot really be addressed in the abstract: we need concrete examples of what we are talking about. Thus, the second reason for attempting to raise some design issues is that the exercise of constructing a model of learning from instruction will provide us with such concrete examples.

Another general question that we can ask about the design of a MOLIM is "Who cares?" Who might benefit from such an exercise? It seems to me to be premature to claim that either instructors or learners (at least as traditionally conceived) could benefit much from thinking about the design of learning models. The payoff, at present, appears to be for the people who fall into the intersection of the categories of instructional designer and cognitive psychologists. The contributors to this volume were selected because of just such a blend of interests and skills. Their answers to some of the questions I have raised are implicit in the work they have presented in previous chapters. Perhaps other "learning engineers" can, in reaching their own answers, begin to apply and direct the kinds of basic research that are required to further our knowledge of both cognition and instruction.

ACKNOWLEDGMENTS

Preparation of this chapter was supported by a grant from the Spencer Foundation. I would like to thank J. R. Hayes, J. A. Jakimik, and D. M. Neves for comments on an earlier draft.