Microcomputer Systems for Authoring, Diagnosis, and Instruction in Rule-Based Subject Matter

Joseph M. Scandura

For many years, I have been heavily involved in basic research directed toward the development of precise and operational, yet comprehensive, theories of complex human behavior. A major outcome of this work has been the specification of constraints which define a class of content and population specific theories, together with various specific examples of what I have called "structural learning theories," which belong to this class. Structural learning theories deal explicitly with the role of content, cognitive processes, and individual differences—and provide a natural scientific foundation for the study and design of instructional systems. Over the past two decades, such theories have been used in the study of rule learning, problem solving, mathematics learning, and the learning of other highly structured subject matters. More recently, we have been moving in other directions as well. For example, we have recently shown how structural learning theories lend operational clarification to the important, but sometimes abstruse, theories of Piaget (Scandura and Scandura, 1980).

In this article, I would like to discuss quite another area, one in which our group has become heavily involved over the past two years—the development of advanced, yet commercially viable, instructional systems for use on microcomputers. Although I have maintained an interest in computer-assisted instruction (CAI) since the 1960s, mainly by virtue of interaction with colleagues such as Karl Zinn, Patrick Suppes, and other pioneers in the field, I avoided direct involvement during that period for three main reasons: (1) The high costs of computers were not only prohibitive for schools, but also for most research (except for a few centers specifically supported for that purpose). (2) Computers of that era lacked reliability. This was often frustrating for people interested in using computers (rather than in computers per se). (3) Most important, from a personal perspective, our understanding of the instructional process at that time was inadequate to provide the solid foundations needed to reliably devise sound CAI systems.

The Microcomputer and Authoring/Driver Systems

Today, with increasingly reliable and low-cost microcomputer systems on the market, the main impediment to widespread use of CAI in elementary and secondary schools is the scarcity of good software and good courseware—good CAI systems. Developing effective and efficient computer-assisted instruction, however, is not an easy nor an inexpensive task. It requires, at a minimum, considerable knowledge of computers, or at least how to program using available computer languages; intimate familiarity with the subject matter involved; and knowledge about how to present the subject matter in ways that are effective for students.

Clearly, simply knowing how to teach a particular subject is not sufficient for developing good CAI. During the past two decades, a considerable amount has been learned about cognitive processes and about how to design effective and efficient instruction based on such knowledge. To date, however, this knowledge has had little impact on CAI.

To be sure, experience and techniques developed in preparing a CAI system for developing one body of content make it easier to develop new systems. But, even under the best of conditions, CAI development, if it is to be educationally sound, is a difficult and time-consuming task—much more complex and time-consuming, for example, than writing a successful text or manual.

To aid the process, considerable attention has been given to the development of general purpose CAI-type programming languages and authoring systems. Among the more widely used CAI programming languages have been Coursewriter, Tutor, MUMPS, etc. These are relatively easy to learn and have been designed to facilitate the selection and presentation of course and test materials to students who interact with the system. While designed to be general purpose, available languages are nonetheless better suited to CAI development in some areas than in others. Moreover, the author is not only free but OBLIGED to specify the sequence of instruction, and these languages provide little guidance in this respect. In effect, although high-level CAI authoring languages facilitate CAI authoring, the process still requires...
considerable familiarity with computer programing, with subject matter, and with instructional design—a combination of knowledge and skills not easy to find in any individual. This fact almost certainly has led as much as anything to the uneven quality of current CAI systems.

In order to ameliorate this problem, a growing number of CAI specialists have argued in favor of developing authoring languages in combination with general purpose "drivers." Thus, authoring systems allow authors to prepare the course material using English and possibly other easily learned codes. This material is usually entered into the authoring system in response to specific prompts. Treating the resulting coded course material as data, the driver program, in turn, operates on the coded material, presenting it in a predetermined, but conditional, sequence determined by the driver and the student inputs (e.g., responses to questions).

In order for authoring/driver systems to function properly, various restrictions must be placed on the data (course material). This is necessary, for example, to insure that the driver is able to locate the right information at the correct time. These restrictions typically limit either the variety of subject matter that can be implemented successfully and/or the instructional effectiveness of the implementation (i.e., while most subject matter can be "forced" into a given format, the resulting instruction may be considerably less than optimal).

In the rapidly expanding microcomputer field, a growing number of companies (including Instructional Micro Systems, a private firm with which the author is affiliated) have recently developed and are beginning to use and/or market what might be called "first generation" authoring/driver systems.

At IMS, this concept of authoring system plus driver is felt to be an extremely promising one, since it minimizes demands on authors and will allow for the development of high-quality, yet cost-effective, CAI for microcomputers. Nonetheless, all currently available (or, to our knowledge, contemplated) authoring systems plus drivers have a major limitation: they do not make serious use of presently available knowledge concerning cognitive processes and instructional systems—knowledge held by the instructional design community but not as yet made widely available to the CAI field.

It is well beyond the commercially viable "state of the art" (i.e., practically speaking, not theoretically) to contemplate, at this time, the development of authoring/driver systems that might be used with arbitrarily complex and interrelated content. There are, however, some kinds of content that are sufficiently well understood to provide considerable hope in this regard. Specifically, research over the previous two decades has given us considerable insight not only into what has sometimes been called rule-based (or "algorithmic") knowledge, but also into effective and efficient techniques for assessing the knowledge of individual students and for providing remedial instruction.

Irrespective of whether all knowledge can be usefully reduced to rule-based or algorithmic terms, it is indisputable that many topics in schools can be so reduced. Ordinary arithmetic algorithms provide a standard, but hardly exhaustive, illustration. Most manipulations of concrete objects, grammar, adding "ing" to verbs, and so on—all lend themselves readily to algorithmic analysis. Relatively complete rule-based analyses of geometry construction problems (Scandura, Durnin, and Wulfeck, 1974; Wulfeck and Scandura, 1977), algebraic proofs (Scandura and Durnin, 1977), geometry proofs (Greeno, 1978; Landa, 1976), and even Piagetian conservation (Scandura and Scandura, 1980) have also been published, along with a large number of compatible analyses in the area of
computer simulation (e.g., Newell and Simon, 1972). Included in these analyses have been rule representations of what have been called higher-order rules, or general cognitive strategies. Analyses along these lines have even been attempted in such relatively abstruse areas as the theory of suppositional proof or natural deduction (Corcoran, 1976) and the foundations of mathematics (Scandura, 1973). While a number of formulations have been proposed as a means of representing algorithmic knowledge, e.g., production systems (Newell and Simon, 1972) and high-level computer languages such as PLANNER (Hewitt, 1972), the present research is based on what Scandura (e.g., 1970, 1977) has called a “rule.” The rule construct plays a central role in the class of structural learning theories (Scandura, 1973, 1977, 1980), which were specifically formulated with the requirements of (i.e., the constraints imposed by) instructional systems in mind and, hence, represent a viable candidate for present purposes.\(^2\)

The Theoretical and Practical Background: Instructional Implications of Structural Learning Theories

Instructional implications of structural learning theories are perhaps best seen by example. Let us consider column subtraction problems as a simple prototype.

In order to utilize structural learning principles in designing instruction, the ESSENTIAL first step is to identify: (1) the educational goals—what the learner is to be able to do after instruction, and (2) prototypic cognitive processes or rules—what the learner must learn if he or she is to perform successfully tasks associated with the educational goals.

In the case of subtraction, for example, let us assume, given a column subtraction problem, that our educational goal is to find the difference. By a prototypic rule, or cognitive process, in this case, I refer essentially to what the learner must master in order to subtract numbers.

“The structural learning theory” provides a general method of analysis, called structural analysis, by which the rules to be learned can be derived from suitably operationalized educational goals. While there are many details still to be completely objectified, the method is relatively systematic and has been applied successfully in analyzing a wide variety of content.

The first step in structural analysis involves selecting a representative sample of problems associated with the goal in question. This sample should be representative in the sense that a person’s being able to solve them would suggest (strongly) an ability to solve arbitrary other problems associated with the goal. In the case of simple subtraction, this might include problems such as:

<table>
<thead>
<tr>
<th></th>
<th>9</th>
<th>879</th>
<th>432</th>
<th>402</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-325</td>
<td>-129</td>
<td>-129</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>??</td>
<td>??</td>
<td>??</td>
<td>??</td>
</tr>
</tbody>
</table>

The second step in instructional analysis involves identifying rules which make it possible to solve each of the selected problems. Identifying such rules involves several identifiable substeps:

(a) Assumptions must be made regarding the MINIMAL ENCODING AND DECODING CAPABILITIES of the students in the target population. In the case of second graders, for example, the teacher/analyst would normally assume that all students are able to distinguish “the minus sign,” the individual digits 0,1,...,9, the columns and the rows, and that all are able to write the individual digits in desired locations. The present illustration builds on this assumption. Consequently, the remainder of the analysis will be inadequate JUST to the extent that these assumptions are in error for students in any given target population.

(b) The analyst must decide the SCOPE OF EACH OF THE REPRESENTATIVE PROBLEMS. This scope effectively defines the domain of the rule associated with the prototype. The problem

<table>
<thead>
<tr>
<th></th>
<th>432</th>
</tr>
</thead>
<tbody>
<tr>
<td>-129</td>
<td>??</td>
</tr>
</tbody>
</table>

for example, might be held prototypic of the entire class of column subtraction problems, namely those formed by varying the individual digits 0,1,...,9, and/or the number of columns. Indeed, in the present case, each of the selected representative problems is prototypic of this same domain. Consequently, in the present case, it is reasonable to assume that there is only one domain, the domain of column subtraction problems.

(c) Next, the analyst must IDENTIFY THE STEPS (operations and decisions) INVOLVED IN SOLVING each of the representative problems. These operations and decisions must be sufficiently simple that using them refers only to abilities that are assumed available to ALL students in the target population (i.e., encoding/decoding capabilities). The operations also must be ATOMIC in the sense that, for each student in the target population, the ability to correctly use an operation once is indicative of uniform success and, conversely, of failure.

The flow diagram in Figure 1 depicts the procedural portion of a rule based on equal additions. In this rule, it is implicitly assumed that
Given this information, the FIRST THING one must do in designing an effective instructional strategy is to DETERMINE WHAT EACH STUDENT ALREADY KNOWS, specifically, that part of what the student knows which is directly relevant to what one wants the student to learn. The process by which this is accomplished has been detailed in the literature (e.g., Scandura, 1971, 1973, 1977), and I will not consider it here. It is sufficient for present purposes to observe that solving a particular subtraction problem involves following one and only one path through the subtraction rule. In effect, there is a unique class of problems associated with each path through the rule. (Note: There are a finite number of paths associated with any given rule.)

A basic principle in structural learning theories is that rules must be represented in terms of operations and decisions that are atomic; they are either totally available or unavailable to any given learner in the target population (i.e., the population the analyst had in mind in performing the structural analysis). The existence of such a representation can always be guaranteed (e.g., Scandura, 1971, 1973; Suppes, 1969).

Collectively, the paths of the product impose a partition (a disjoint, but exhaustive, division) of the domain of column subtraction problems into types, or equivalence classes, of problems. For each path, there is an associated type and vice versa. In effect, success or failure on any one problem associated with a class of problems provides complete information as to the availability to the student of the corresponding path. For example, in

![Diagram of the Equal Additions Algorithm for Subtraction](image)

the problem is solved by following the path defined by operation one, then two, then three, and back to two and three, then two and three again, before stopping.

By testing on a small, finite set of problems, it is possible to identify precisely and unambiguously which parts of the subtraction rule any given student knows and which parts the student does not know. Such testing, in effect, defines the student’s entering level: it determines which paths the student knows and which he or she does not know.

A significant amount of supportive data has been accumulated over the past two decades, with respect to a wide variety of mathematical tasks, with subjects ranging from preschoolers to Ph.D. candidates. Given a class of tasks, the general form of each study went as follows:
1. One or more rules were identified which were adequate for generating solutions to each of the tasks and compatible with the way a knowledgeable or idealized member of the target population might be expected to solve them.

2. These rules singly and/or collectively applied were used to partition the class of tasks into equivalence classes.

3. Subjects in the target population were tested on two items (tasks) from each equivalence class (type of item).

4. Performance on an item from each equivalence class was used as a basis for predicting success or failure on the other (second) item.

With highly structured tasks run under carefully prescribed laboratory conditions, and given performance on initial items, it has been possible to predict performance on new (second) items with over 96 percent accuracy (Scandura, 1970, 1973; Scandura and Durnin, 1978). When testing took place under ordinary classroom conditions, with the subjects run as a group, the predictions were accurate in about 84 percent of the cases (Durnin and Scandura, 1973).

In the latter study, which involved subtraction directly, the equivalence classes determined via the underlying rules were compared with the item forms (types of items) identified by Hively et al. (1968) and Ferguson (1969). Whereas the levels of prediction on success items were approximately the same, the rule-based approach was more reliable and more efficient (requiring only half as many test items). Moreover, there is a hierarchical relationship among the paths of any rule (with paths lower in the hierarchy being included in those at higher levels).

Such hierarchies provide a theoretically derived and empirically verified partial ordering (hierarchy) of selected test items, according to difficulty. This difficulty hierarchy has been utilized to provide even more efficient assessment. Once a student has failed at a given level, for example, there is no need to test on more difficult tasks in the hierarchy. Conversely, there is no need to test on tasks at levels below where the student has already succeeded. In this regard, more can be said about such things as testing in situations where more than one rule is involved and about increasing efficiency via sequential testing (e.g., Scandura, 1971, 1973, 1977), but this is not necessary for present purposes.4

PRESCRIBING INSTRUCTION, then, follows directly from what the student knows. All one needs to do is to IDENTIFY THE MISSING PORTIONS OF THE DESIRED SUBTRACTION

RULE AND TO PRESENT THEM TO THE STUDENT. The theory is neutral on whether this information should be presented, say, in an expository or a discovery manner. Thus, for example, deciding on the appropriate method of presentation depends on secondary objectives that the instructional designer may (or may not) have in mind (e.g., to help students learn how to detect regularities). The important part insofar as being able to perform subtraction is concerned is simply to be able to perform according to the rule.

As an illustration, suppose a student’s knowledge may be represented by the flow diagram shown earlier, minus only the loop involving operation five (add ten to the top and subtract, go to the next column and add one to the bottom). In this case, a computerized instructional system would need only to make sure that the student knows, at the appropriate points, how to add ten to the top number in the column and how to add one to the bottom of the next column. Where the student knows less, of course, one would start with the simpler prototypes (partial rules representing what the student knows) and gradually “elaborate” (cf. Merrill, 1980), or add increasing detail until the student has mastered the entire rule.

To summarize, I must emphasize that this illustration of prescriptive aspects of the structural learning theories constitutes only a simple prototype. It “epitomizes” the instructional aspects of the theory. The theoretical system itself provides a far more generalized basis for instructional prescription—which in principle, may be used with any subject matter (or educational goal) that might be of interest.

Relating Structural Learning to Computer-Based Instruction

In the present context, the basic question is how current understanding of rule-based analysis, diagnosis, and learning might be utilized in the development of improved authoring systems and general purpose drivers. The first thing to realize in this regard is that authoring systems correspond to
"structural analysis" and that general purpose drivers correspond directly to the diagnostic and remedial (instructional) aspects of the structural learning theory. We emphasize the term "structural learning theory" in this context, rather than the plural "theories," because both structural analysis and the relevant test (diagnostic) and instructional aspects of the theory are totally independent of content (and student population).

To convert structural analysis into a computerized authoring system, of course, would be a major undertaking. It would amount essentially to developing a theory for developing (rule-based) theories. Basic research in this extremely important area is proceeding apace, with generous support from the Spencer Foundation, but it is too soon to say how much more objective and systematic structural analysis can be made, much less whether the process can be computerized.

The situation with regard to diagnosis and remediation is more straightforward, since almost all of the requisite basic knowledge has been available for some time. Hopefully, practical solutions to this aspect of the problem may become available in the near future.5

Conclusion

I have argued that microcomputers provide a promising new tool which have the potential of revolutionizing the way in which routine, rule-based types of instruction are delivered. Microcomputers are inexpensive, reliable, and sufficiently powerful to meet many important educational demands.

Rather than using general purpose CAI-type languages, however, I have suggested that high-quality progress can be made more quickly and economically by greater use of authoring/driver systems. Specifically, we have argued that such systems should make far greater use of what is currently known about cognitive processes, and particularly about structural learning.

I have outlined only a small portion of what is already known about the design of effective and efficient instructional systems. What has been presented here applies only where to-be-learned competence can be expressed in terms of single rules, or algorithms. Indeed, most of the work on which these remarks are based is at least ten years old. It is, one would think, about time the field of CAI began to capitalize on it.

Under the auspices of a new, privately funded company, Instructional Micro Systems, we have developed several authoring/driver systems which approximate some of the features discussed. Recently completed was a Critical Reading System designed to teach students to apply basic logical inference rules using written material. Empirical evaluation of the materials used in this system has indicated its effectiveness as an instructional tool. General principles of structural analysis also were used in developing a College Entrance Examination Preparation System. This program is designed to teach and reinforce the verbal and mathematical skills which students will encounter in taking the College Board examinations. Both systems are now being marketed by Borg-Warner Educational Systems under the label "MicroSystem80." Planned development draws even more heavily on these ideas. Currently, we are developing highly sophisticated MicroSystem80 systems in basic mathematical skills and language arts.

In conclusion, I would like to caution that there is a large gap between what we know is possible theoretically and what is commercially and practically viable. If one were to fully utilize what has been learned in structural learning over the past two decades, it would be possible to deal with far more sophisticated content than that involving only single rules. Sophisticated applications of this sort may become more feasible with the next generation of computer hardware, but I suspect that the more limited type of application discussed above will push present microcomputers about as far as they can go.

Notes

1. This research is supported in part by a grant to the author from the Spencer Foundation and in part by Instructional Micro Systems.

2. Formally speaking, a rule is defined as a triple, consisting of a domain, a range, and a restricted type of procedure (algorithm which may or may not terminate) operating on the domain.

3. If operation two of the subtraction rule (i.e., subtract bottom number from top number using facts...) were not atomic, for example, which it probably would not be for a group of first graders, one might break down the operation into a more detailed subprocedure that explicitly represented each subtraction fact. In general, the less sophisticated a group of students, the more detail is required. Conversely, the more sophisticated the students, the larger the components may be. Hence, complexity of a rule representation is a relative notion, which does not depend solely on the apparent task complexity.

4. In addition to my own work in this area, it is important to observe that an increasing amount of similar research is being generated by others and with similar positive results. Thus, for example, Klair (1978) has used similar assessment procedures to ascertain developmental level with respect to the balance beam task. Brown and Burton (1978) extended aspects of this work, first by identifying what Scandura (in Spada and Kempf, 1977) has called "error" rules, or what Brown and
Burton call "bugs"—incorrect ways of performing given tasks, and second by implementing the diagnostic method on their computer.

While it may be interesting, and useful for some purposes, to know what (error) rules are responsible for a students' incorrect answer, this information may not be necessary for instructional purposes—where atomicity assumptions hold and where the goal is to get the student to perform correctly. Indeed, my earlier work on rule selection (e.g., Scandura, 1971, 1973) suggests that giving extra attention to erroneous procedures during the instructional process, presumably along with the correct way of doing things, might well lead to subsequent confusion on the part of the student. "Of the two things the teacher talked about, which am I supposed to do?"

S. Indeed, a master's degree student in another department of the University of Pennsylvania attempted to implement a diagnostic and tutoring system along these lines during 1973-74, but inadequate time, computer expertise, and understanding of the basic theory resulted in only partial success.

References


Forthcoming Articles

Among the articles scheduled to be published in the February issue of Educational Technology are the following:

- The Teacher Factor in the Supply and Demand Curve for Technology in the Schools. By M. Vere DeVault and John D. Chapin.
- Photonovels and Comics as Instructional Technology. By Fred B. Lindstrom, Naomi Lindstrom, and L.L. Johnson.