Theoretical Foundations of Instruction: Past, Present and Future

JOSEPH M. SCANDURA
University of Pennsylvania, and Scandura Intelligent Systems, Inc.

(Received June 1994)

Computer technology has undergone major advances in recent years. Important advances also have been made in understanding both complex human learning and the instructional process. Instructional design methodologies too have evolved in a positive direction. Nonetheless, instructional design still clings largely to its taxonomic roots. Different approaches are typically used to achieve different learning outcomes—one method for facts, say, and another for teaching problem solving.

I believe the time has come in instructional design for renewed emphasis on instructional theory. As Franz Schott said at a NATO ASI (Tennyson, et al., July, 1993, Grimstad, Norway), “We have not in recent years learned much that is new about instructional theory.” I would add that we have learned even less that is new about fundamental theory. By this, I do not mean ongoing refinement of classification schemes in the sense of Bloom’s taxonomy, Gagné’s Conditions of Learning, or the 13 or so kinds of learning supported by Dave Merrill’s authoring system. Nor, do I mean content-specific classifications of knowledge such as the one I published on mathematics in 1971, or the current vogue in constructive thinking.

I believe that instructional theory should be practical. But I am of the old school insofar as theory is concerned. Good instructional theory is not a compendium of diverse theoretical perspectives, not is it constructive philosophy or any set of techniques or methods. Instructional theory should strive for comprehensiveness, internal consistency, parsimony, precision, and it must be operational.

Do we have such a theory today? Not entirely, but I do believe that the Structural Learning Theory (SLT) provides a sound foundation on which to build. Major developments are summarized in Figure 1.

In this article, I will not dwell on theoretical details—nor is that necessary since much of the material is available in published form. Instead I hope to summarize some major concerns and to convey both a better understanding of instructional theory and the advantages that such theory offers for the future of instructional design.

---

1 I would, however, call attention to the evolution of notation in these writings. In the 1960’s and early 1970’s emphasis was on the basic mathematical notions of sets and functions. Data and process abstraction have played a more direct role in recent writings, particularly in my recent work in software engineering.
Figure 1. Overview of major phases of Scandura's research arranged chronologically. Captions in the rectangles refer to key results. The numbers within brackets correspond to items in Scandura's list of publications. Relationships are represented by arrows between the rectangles.
Background.—My research on instructional problems has in the past often been characterized as "nontraditional". Indeed, my goals have been different from those of many educational researchers—so it may be helpful to provide some sense of what led to my current thinking on instructional theory. My major goals in instructional theory, as in most other forms of personal understanding, have involved finding commonality—single sets of assumptions sufficient in principle, to explain, predict and/or control the relevant phenomena.

Even as a three year old experiencing my first thunderstorm, I don't remember being frightened as much as I was puzzled. Where did all that noise come from? When asked, my mother thought for a moment and replied that it was God in heaven. Needless to say, this didn't ease my bewilderment. My parents were having a house built at the time, however, and I do remember hearing the sound of workmen rolling large barrels over a rough wooden floor. It occurred to me that thunder might be caused by God rolling barrels on some floor up in heaven. Unfortunately, I couldn't figure out what might hold up the floor for God to roll his barrels on.

It would be stretching the point to say that my next step was to fly a kite in a thunderstorm. Nonetheless, for as long as I can remember, I've always been interested in finding commonality among seemingly disparate information. More specifically, in studying mathematics as a teenager, I became increasingly interested in the thought processes involved, almost to the point where the mathematics itself became incidental. My curiosity increased gradually, but it wasn't until working on my Ph. D. dissertation in mathematics and education that I became absorbed in the problem. At that time, in the early 1960's, my committee members fell in two diametrically opposed camps: (a) Psychologists and educational researchers proposing the rigorous experimental but often sterile research methods of the early 1960's and (b) mathematicians, the content-centered, more intuitive "action research" associated with the "new math". The gap between the two groups was huge—a gap that I increasingly found myself trying to fill.

My dissertation was an attempt to reconcile hard experimental evidence with the claims being made for discovery learning by new math pioneers. Experimental data and tape recorded analyses of the teaching-learning process were used in conjunction with an artificial body of content or abstract card tasks (Ref. #2) I had developed to parallel the new math and to minimize individual differences at the beginning of instruction (Ref. #3's 3, 11, 12, 13, 14, 19). Neither set of advisors was especially happy with what I found. The timing of information—when information is given—relative to what individual students know at the time is far more important in determining learning outcomes than whether the instruction itself is given in an expository manner or by guided discovery (Ref. #3.).

This result raised the question of how one really knows what students know. More precisely, how can one represent what people know in a way that has behavioral (i.e., observable) relevance? Behavioristic theories prevalent at the time attempted to reduce all human capability to S-R associations. Precognitive formulations during this period were hopelessly vague. There was much in these proposals that I did not find satisfactory. Yes, on introspection, we all find our thoughts skipping from one idea to another and in that sense, one might say that there are associations between them. Nonetheless, I felt there was a natural order of things cognitive, regularities which govern everything
from complex problem solving to potential links between thunder, say, and rolling barrels on a wooden floor.

In this context, recall Robert Gagné’s original edition of the *Conditions of Learning.* Gagné’s goal was to build a bridge of sorts between S-R behaviorism and the more complex kinds of learning characterizing human thought. He explained concepts, for example, in terms of S-R associations in which there are many stimuli and a single response. Rules, or what he originally called principles, were defined as chains of concepts.

In effect, Gagné accepted the S-R association as the fundamental unit of analysis and attempted to represent more complex forms of knowledge in these terms. As I showed in a series of analyses (Ref. #30, B4) however, rules do not lend themselves well to representation in terms of S-R associations. For each input to a rule there is a unique output and there may be any number of such inputs. Every quadratic equation, for example, has a unique pair of solutions. Gagné’s original formulation of rules, as chains of concepts, allowed for any number of inputs and outputs—but the connection between them was not necessarily unique.

Associations and concepts, on the other hand, are easily represented in terms of rules. Consequently, it seemed that the study of cognition might benefit from a different perspective. Rather than viewing concepts and rules as complexes of associations, why not take the rule as the basic unit of analysis and view concepts and associations as special cases (Ref. #’s 30, B4).

In addition to parsimony, this approach also made behavior highly predictable. Performance on one instance of a rule, for example, made it possible to predict the behavior of individual students on new instances of the rule with an unusually high degree of precision. Successful performance on one problem in a class invariably implies success on other problems in the class, and conversely for failure (Ref. #’s 30, 50).

Although this seemed to be a step in the right direction, it did not get at perhaps the most important question. How do people use existing knowledge to derive new knowledge? It was this problem that led me to reformulate the notion of higher order rules. Gagné recall, had originally defined higher order rules or principles, as chains or combinations of rules. By way of contrast, I defined higher order rules as rules which act on and/or generate other rules (Ref. #’s 55, 67, 70, B3). Rather than representing composite knowledge as proposed by Gagné, higher order rules—as defined in the SLT (Ref. #’s 55, B3)—represent (higher order) knowledge for creating new (e.g., composite) knowledge from existing knowledge. Given a novel problem for which one does not already know a solution rule, for example, one can use higher order rules to derive or “figure out” rules (i.e., solution procedures) for solving the problem. One type of higher order rule is exemplified by trial and error behavior (for example, by the chicken in a Psychology I textbook eventually finding its way to food behind a barrier. Humans, of course, are capable as well of more sophisticated methods.

**Behavior and Cognition.**—The basic datum in studying human cognition is human behavior. As educators and psychologists we do not have the luxury of looking inside the brain. Although there is unquestionably a good deal to learn about the brain, I

---

daresay that we will never be able to teach higher mathematics or poetry, say, by programming human grey matter. Of necessity, we work at a higher level.

Direct manipulation of the brain is effectively impossible with respect to school learning, even at the physiological level. To see why, one has only to consider the complexities involved in programming and/or debugging complex computer software. Then, consider the additional complexities of manipulation at the binary (or in the case of the brain at the neuronal) level. Some problems can be downright intractable, even in modern high level programming languages. Finally, recall that the human brain is infinitely more complex than the most complex computer program seriously contemplated.

These observations aside, I do believe that quasi-physiological models of the brain have considerable potential. Training so-called “neural networks” by example can play a useful role in learning classifications. Complex human learning however, involves complex processes as well. Such processes involve action and not just decision marking.

In short, we cannot hope to get the information we need as instructional designers by simply studying the brain. Nor can we directly manipulate the brain of a child. We can only infer things about the mind by observing behavior.

That, of course, is the problem. Exactly how do we find out what a student does and does not know by observing behavior? Traditional cognitive science tackles this problem by building models—mini-theories if you will—about how people solve this or that problem, or perform this or that task. These models, in turn, lead to behavioral predictions. Often, these predictions map reasonably well onto behavior. But they typically say little about individual behavior. Although consistent with behavior averaged over groups of people, these models may differ significantly from what individuals do and do not know. To address individual behavior, one might require as many models as there are individuals, clearly an impractical solution. To make matters worse, different theories are needed for different content domains.

**Structural Analysis.**—Clearly, fundamental instructional theory requires a different approach (Ref. #’s B1, B3, 70). Rather than developing each content-specific theory de novo, a common method, meta theory if you will, is needed for constructing content-specific theories. Task analysis provides a case in point. It has been used to analyze a wide variety of task domains. Sharing much in common with structured analysis in software engineering, task analysis identifies the ingredients (prerequisites) for performance at successive levels of abstraction. What it does not do however, is specify cognitive processes sufficient for actually solving problems in the domain.

Structural analysis is a form of cognitive task analysis which explicitly provides for higher as well as lower order knowledge (Ref. #’s 70, B3, 131). Given an arbitrary content domain, no matter how simple or how complex, structural analysis results in a set of rules and higher order rules which collectively make it possible to solve a wide variety of tasks in the domain, including novel tasks not directly considered in the analysis.

Structural analysis provides a different approach to the question of individual knowledge, one based on the fact that we can never know *everything* that any person knows. Nor is that necessary. We need only know what an individual knows relative
to what we are interested in observing. Furthermore, what we are interested in observing can always be represented in terms of rules and higher order rules identified via structural analysis. These rules provide a prototype, a measuring device of sorts, against which to measure the knowledge of individuals.

In effect, although we can never know exactly what a person knows, we can determine which parts of which rules are and are not known. Moreover, extensive research shows that this information can be used to predict individual behavior with an unusually high degree of accuracy.

**Some misunderstandings about Structural Analysis.**—Let me emphasize a few points about structural analysis that are often misunderstood.

1. Structural analysis is strictly cumulative. Suppose one enlarges the domain of analysis: for example, by extending an analysis of “ing” endings to a broader range of grammar, or by generalizing computational arithmetic to incorporate concrete meanings as well. Do we have to “throw away” the initial results of structural analysis and start anew? Must we start over, by applying structural analysis to the enlarged domain? Happily, the answer is no. Structural analysis simple begins where the previous analysis leaves off. (Ref. #’s B3 [esp. Chapters 3, 13, 14]).

2. Structural analysis is applicable to essentially any content, no matter how broad and comprehensive (Ref. #’s B3, B6, 113) or how narrow and focused (Ref. #’s 113, B8). Ironically, full analyses of both kinds of domains invariably result in higher order rules. In the case of broad domains, the higher order rules typically serve to create (i.e., generate) entirely new rules (Ref. #’s B3, B6). In the case of highly specific knowledge, the higher order rules commonly involve automatization (Ref. #’s B8, 113). These higher order rules replace procedural complexity with structural complexity and consequently, result in more efficient rules.

3. Structural analysis may be applied to a given content domain from any number of different perspectives. Each perspective corresponds to a different paradigm for addressing that content. The borrowing and equal additions methods of column subtraction provide a simple example (Ref. #63). Pragmatic versus theory-based approaches to instructional design provide others. Yes, instructional design itself can be subjected to structural analysis.

   Each perspective corresponds to a common view shared by individuals in some group or subgroup. The rules associated with each such perspective best characterize the knowledge of individuals in the corresponding group.

   This observation raises a variety of research questions. Communication, for example, may be difficult if not impossible where teacher and learner—or two instructional designers for that matter—approach a problem domain from different perspectives. To communicate, the two must first establish common ground. This requires common elements in the results of structural analysis from the opposing perspectives. This is an area pregnant with research opportunities.

4. It is not essential that the results of structural analysis capture individual knowledge exactly. It is sufficient only that the identified rules yield (equivalent)
predictions. Given a problem domain, there always exists some level of analysis beyond which predictions cannot be distinguished. If accurate computation is all that matters from an instructional design perspective, for example, a more detailed analysis involving processing time would be superfluous. The former analysis might be quite sufficient. Having said this, I would add a caveat. Just because an analysis (i.e., rule set) cannot be distinguished within a defined domain does not mean that it cannot be distinguished. Generalizing or enhancing a domain (e.g. by introducing response times), for example, makes it possible to distinguish behaviorally between different levels of analysis (Ref. #’s B3, B6).

(5) Detailed structural analysis of non-trivial domains can take a lot of work. It is not essential, however, to detail everything. Experience shows that even crude structural analyses can have valuable predictive benefits (Ref. #’s B6, B8). Although, determinstic (actually non-deterministic) predictions are theoretically possible, this is not essential. The question is whether structural analysis, formal or otherwise, adds a degree of predictability that would not otherwise have been possible.3

Rules.—Since rules play a central role in the Structural Learning Theory (SLT), let me be just a bit more precise on what they are. All rules have a domain, a range, and an operation or process connecting the two. The domain may consist of any number of elements, and similarly for the range. The range of a rule, however, is not necessarily what one gets by applying the operation to elements in the domain. Rather, the range of a rule defines the expectations of applying the rule. Rule domains and ranges, together with a universal content-independent control mechanism, play a central role in determining which rules will be used in given problem situations (Ref. #’s 55, 113, B3).

Higher order rules are simply rules in which the elements of the domain and/or range may themselves be rules. All higher order rules combine, or otherwise modify, given rules in one way or another. Nonetheless, it is often convenient to distinguish two kinds of higher order rules. Some higher order rules produce new knowledge, for example generalizations of existing rules or new combinations of such rules (Ref. #’s 55, 67, B3, B6). These are the kinds of rules that are used in learning or problem solving situations where the student does not already know how to deal with given problems.

Other higher order rules, however, result in automization. Such higher order rules make existing rules more efficient (Ref. #’s B8, 113). They convert operational steps in the procedural portion of a rule into automatic processes encapsulated in domain or range objects. The terms “objects” and “encapsulation” are used purposefully in this context to emphasize a very strong analogy between what I am proposing and the object oriented paradigm currently in vogue in the software engineering community (Ref. #’s 162, 164). When applied to the design of software systems, structural analysis constitutes a new cognitive paradigm which effectively generalizes and incorporates the traditional object-oriented paradigm. But let me not diverge.

---

3 This does not necessarily imply a need for traditional statistical methods. Indeed, so-called “fuzzy sets” could well provide a better model where content does not readily lend itself to sharp partitioning.
Problem Solving.—All this is well and good—but it still does not deal with the question of how to account for creative or problem solving behavior. One possible solution derives from a very simple idea. Recall the chicken in your Psychology I textbook placed in front of a glass barrier separating it from its food. First, the chicken bangs into the glass. After crashing enough times, the chicken by trial and error may eventually find its way around the barrier. While I'm not inclined to attribute high intelligence to the typical hen, or rooster for that matter, trial and error is just the name given to a class of higher order processes for deriving solutions for unfamiliar problems. Indeed, such behavior appears universal. When faced with a problem for which one does not have a ready made solution process, humans—and apparently chickens as well—set about the task of finding a solution that will work (Ref. #1's 55, 67, B3, B6).

When thus formulated, what I earlier called the "goal switching" mechanism will not work with arbitrary knowledge in a generalizable way. In its original form, goal switching subtly but necessarily required content-specific knowledge in addition to the mechanism itself (Ref. #1's B6, 113). There is however, a solution to the problem, one which based on my other experiences with the U.S. Patent Office appears to be patentable. In short, there is far more to this simple minded idea than first meets the eye.

Overview of Instructional Theory.—Let me now turn specifically to instructional theory (Ref. #1's 115, 141, 143, 146, 148). Notice in Figure 2 that the teacher and learner necessarily communicate through—or with respect to—one or more perspectives on a body of content. These perspectives consist of sets of rules, some of which may operate on other rules.

The teacher normally is assumed to have full knowledge of the rules and higher order rules constituting the target content domain. Only the learner's knowledge is at question. This knowledge is assessed relative to rules in the content domain (Ref. #1's 146, 148). Theoretical problems become more complex when both teacher and learner have one partial knowledge.

Time does not allow detailing exactly how the teacher decides what problems to present to the learner, how the teacher assesses what the learner knows and/or how the teacher determines what information to present to the learner at each point in time. For this, I refer you to the listed publications (Ref. #1's 146, 148).

I would, however, emphasize one important point. This teaching-learning paradigm is not only applicable to procedural knowledge. The theory applies to all kinds of knowledge whether that involves learning simple facts, learning how to apply general purpose solution rules (such as those used in solving subtraction problems), or using existing knowledge (rules) to derive new rules for solving novel problems.

Based on earlier research, it is fairly clear how the latter two forms of learning can be handled. On deeper analysis, one also finds that learning simple facts follows the same principles. What may appear to be simple facts, almost always have a complex genesis. In our analyses of Piagetian conservation, for example, what appear to be automatic conserver responses are derived from more basic knowledge via automization (Ref. # B8). Even S-R learning, involving well-established and familiar stimulus and response objects, follows the same pattern once context is taken into account. In traditional S-R research, for example, each S-R association is embedded in a list ("situational cognition" perhaps?).
Implications for Instructional Design.—Suppose for a moment that we could develop a general purpose intelligent tutoring system, and that this tutoring system could accommodate essentially any type of subject matter (Ref. #’s 146, 148). Unlike current tutoring systems, the system I have in mind would automatically generate test problems as needed. It would automatically infer what the learner did and did not know. And it would provide precisely that instruction each learner needed when it was needed. It is important to distinguish this type of tutoring system—both from traditional authoring systems, which although “general purpose” require designer input for instructional specifications as well as content, and from intelligent tutoring systems, which although “intelligent” find content and instructional decision making inextricably intertwined.

Imagine next what happens as the teacher knows less and less about the content to be taught. The interaction gradually degrades. Eventually, the teacher may perform
comparably to the learner. In this case, we have a model for cooperative student learning.

Next, suppose that the teacher knows and contributes nothing to the dialog. Here, the learner is interacting with an essentially random environment centered on the content domain. The learner finds himself exploring the content in an open-ended simulation environment.

In all of these cases, our general purpose tutor would relieve us of a good deal of work. The construction of test items, their answers and instruction, as well as the sequencing of testing and instruction would all be handled automatically. In the case of simulation, of course, the learner alone would be the focus of interactivity.

In order to work as a full fledged system, the general purpose tutor needs content. More particularly, it needs sets of rules which characterize the problem domain in question. This requires structural analysis. Indeed, in this context, instructional design actually reduces to structural analysis. Once built, the tutor would take care of everything else: devising test problems, sequencing of tests and instruction, even deciding how information is to be presented.

Structural analysis, of course, does require thought and effort. The tutoring system, however, would release you, as an instructional designer, of essentially all instructional decisions. These would be fully automated. In comparison with traditional authoring systems (whether developed by us, e.g., S-6, or others), the savings would be considerable. There would be no need to worry about instructional decision making—about which or how many test items to use to assess student knowledge or about what instruction to give and when. Moreover, the results would be far more predictable. If we have done our job right—our goal would be to optimize the teaching-learning process—student learning would be far more efficient than if instruction were designed by a human. In this context, our analyses suggest that human teachers simply could not attend to the detail required (for optimized instruction) (ref. #’s 146, 148).

Of course, I am not so naive as to believe that we can easily develop such a tutoring system—one that will accommodate all varieties of content. Having supervised the development of some very large (million-line plus) systems in software engineering, I am well aware of the technical issues that will have to be resolved. But, that is where theory comes in. Sound instructional theory can provide a beacon guiding development. It can lead to powerful, highly efficient and truly general purpose intelligent tutoring systems. It can ensure development that is cumulative rather than haphazard. More generally, sound theory can save money and a lot of wasted effort.

We should begin.

References


70. Scandura, J. M. Durnin, J. H. and Wulfek, W. H., II. Higher-order rule characterization of heuristics for compass and straight-edge constructions in geometry. Artificial Intelligence, 1974, 5, 149–183. (Also in Problem Solving; see B-6.)


113. Scandura, J. M. Problem solving in schools and beyond: Transitions from the naive to the neophyte to the master. *Educational Psychologist*, 1981, 16, 139–150.


S-1. Critical Reading, Fall 1980 (8 disk system).


S-6. SURPAS Authoring system with "intelligent" interface, Spring 1983.


