System Issues in Problem Solving Research

JOSEPH M. SCANDURA

University of Pennsylvania

This article raises the fundamental issue of whether various aspects of problem solving can reasonably be studied as isolated phenomena. Special attention is given to knowledge representation, the process of analyzing problem domains, individual differences, learning and "extra-domain" knowledge. Among other things, it is argued that: (a) system requirements (making provision for individual differences: measurement, learning, etc.) impose inviolable constraints on knowledge representation, (b) problem domains may be analyzed systematically, and (c) "extra-domain" knowledge is a misnomer, and that all knowledge is best viewed as relativistic. More generally, it is concluded that problem solving is best conceptualized as a rigorously defined, yet unified theoretical system.

The major goal of my introductory remarks is to raise a fundamental issue in problem solving research — specifically the issue of whether various aspects of problem solving should be studied as isolated phenomenon — or, whether problem solving research might better be viewed within an equally precise, yet broader systems perspective.

The goal of most problem solving research back in the late 1960's and early 1970's was to uncover general relationships involving such variables as aptitude, problem difficulty, meaningfulness, problem solving styles, etc. By way of contrast, those working in structural learning (e.g., Scandura 1971, 1973, 1977) and various other schools had long come to the conclusion that any reasonable approach to problem solving research would have to deal directly with specific knowledge. The various knowledge-specific schools differ in other respects, of course.
SOME ESSENTIALS OF STRUCTURAL LEARNING THEORIES

As a base for comparison, let me briefly review some essentials of the Structural Learning Theory (e.g., Scandura, 1971, 1973, 1977) — not only for the obvious reason that structural learning has been the main focus of my work — but also because it appears to be unique in its overall concern for human problem solving as an integrated system.

1. Whereas all cognitive theories recognize a distinction between domain specific knowledge and control, this distinction in structural learning theories is sharp and fixed. The former refers to knowledge associated directly or indirectly with given problem domains; the latter, to such things as a control mechanism governing the use of domain specific knowledge, processing capacity and speed.

In the case of control, for example, we discovered almost 15 years ago now that human problem solvers come "wired in" with a goal-switching mechanism (e.g., Scandura, 1971, 1973, 1974). It need not be taught or learned because humans behave as if it is already there.

2. A second major problem has been how to deal with individual differences in problem solving. One alternative has been to employ normative testing as in most correlational studies. Another, as in computer simulation, has been to devise theories which directly reflect individual differences.

By way of contrast, structural learning theories make a sharp distinction between prototypic competence, or idealized knowledge (associated with given problem domains), and relevant individual knowledge (e.g., Scandura, 1971, 1973, 1977). Prototypic knowledge serves as a standard against which individual knowledge is measured. Assessed in this manner, individual knowledge has been shown empirically to provide a highly reliable basis for predicting problem solving behavior. The behavior of individual problem solvers on specific problems has been predicted with 80–99% reliability depending on the test conditions (e.g., Durnin & Scandura, 1973; Levine, 1966, Scandura, 1967, 1970, 1973, 1977).

3. Given the importance of prototypic knowledge, considerable attention also has been given to methods for identifying prototypic knowledge. Originally, task analysis was concerned with behavioral requirements (e.g., Gagne, 1962.) Later, in recognition of major differences between behavior and the cognitive processes underlying such behavior, the method of structural analysis evolved over the past 15 years (e.g., Ehrenpreis & Scandura, 1973; Scandura, 1971,
1971b; Scandura, Durnin & Wulfek, 1974). What distinguishes structural analysis from other approaches to cognitive task analysis is the explicit attention given to higher order knowledge or rules. Higher order rules operate on and/or generate new rules and have been shown to provide an explicit basis for explaining, predicting and even controlling problem solving creativity.

COMMONALITIES WITH OTHER MODERN PROBLEM SOLVING THEORIES

Since the late 1960's and early 1970's when the Structural Learning Theory was originally proposed, many aspects of the theory have become commonplace. Basic ideas have been adopted, implicitly or explicitly, to some degree in almost all modern problem solving theories.

1. Not only is it generally agreed today that domain specific knowledge is the most important factor in problem solving behavior, but the term "rule" itself has been widely adopted, albeit often in modified form.

2. Although details differ, the goal-switching mechanism also is coming into acceptance. In Artificial Intelligence, for example, it has become increasingly common to combine both data and goal driven inferencing as is done in goal-switching.

3. A growing number of investigators (e.g., Seigler, 1976; Holzman, Glaser & Pellegrino, 1976) also have adopted prototypes as a means of assessing individual capabilities.

4. There is broad consensus on the need for ways to efficiently conduct knowledge analysis. In education, the process has been quasistystematized and goes under the label of task analysis. In artificial intelligence, the term "knowledge engineer" has been coined to refer to cognitive scientists skilled in the analysis of problem domains.

Rather than dwell on similarities, however, or specific variations, I wish to emphasize just a few fundamental issues on which there is little agreement.

TYPE OF REPRESENTATION

Almost everyone agrees on the need for precise representation of
knowledge. Nonetheless, many believe that the type of representation, whether declarative, procedural, production system, or whatever, is primarily a matter of taste and preference. Those who deify mathematical formalism, for example, frequently give primary credence to Post-type production systems (productions are often called “rules”). Others prefer relational networks; and still others, sets or procedures.

In one important sense, the above kinds of representations are all alike. Human problem solving behavior can equally well be represented in any of these forms.

Recall, however, that different programming languages are often preferred in different applications. In a similar manner, human problem solving imposes severe constraints on how knowledge might best be represented. For example: (a) The representation chosen certainly should allow for assessing individual knowledge: One would want a representation which provides a convenient basis for representing individual differences as well as prototypic knowledge. (b) Similarly, it should be possible to represent knowledge at arbitrary levels of detail (the desired level of prediction or explanation of problem solving behavior may vary greatly). (c) And it should provide naturally for the acquisition of knowledge. (d) Finally, a useful form of representation should facilitate the process of representation itself. So much the better if knowledge representations underlying given problem domains can be derived systematically.

Neither space nor time allow complete analysis of the various major forms of representation with respect to all relevant criteria. Let me just illustrate by listing some limitations associated with the widely used production systems form of representation.

First, even individuals who have adopted the production system formalism for most of their work use a quite different formalism, based on simple decision trees, in assessing behavior potential (e.g., Seigler, 1976; Klahr, 1978). Decision trees are just a specific case of structural learning rules.

Second, productions are by definition the basic elements in any production system (used to represent knowledge). To represent the knowledge in more detail requires replacing individual productions with production systems — thus quickly yielding production systems whose elements are production systems whose . . . . This might be okay if production systems provided a convenient basis for representing individual knowledge — but they do not.
Third, for many years, production systems were used almost exclusively to represent performance during problem solving. Little attention was given to knowledge acquisition, motivation or any number of related psychological processes. Indeed, learning was thought to be extremely complex and difficult to study and for the most part was avoided (e.g., Newell & Simon, 1972). Even today, learning studies typically involve the acquisition of individual productions rather than production systems per se (e.g., Waterman, 1974). Since nontrivial problem solving behavior is explained in terms of production systems, the process of learning viewed from the production systems perspective must necessarily deal with production systems per se and not just individual productions. Consequently, explanations of learning have been both less general and far more complex than they need to be.

Fourth, systematic methods (e.g., Scandura, Durnin & Wulfeck, 1974; Scandura, 1982, 1984a, 1984b; also see “structural analysis” below) already exist for identifying the prototypic rules underlying given problem domains. To date, I am unaware of any conclusive research in this direction with regard to production systems. This is not to say that it might not be possible to do so — especially since adapting an existing approach is almost always easier than creating one in the first place. The basic question will be how such a method will compare with structural analysis as regards such things as cohesiveness, parsimony and ease of use.

In structural learning theories, all knowledge is represented in terms of rules. Like productions, rules have a domain and an operation. They are unlike productions, however, in two important ways:

(1) Neither rule domains (structures) nor the operations (procedures) need be atomistic. In general, both are themselves composed of atomic elements — hierarchically arranged conditions or equivalence classes in the case of domains, and procedures in the case of operations. (Note: For reasons beyond the scope of this discussion, rule procedures may not be recursive, e.g., see Scandura, 1981.)

It also is worth noting that there is a duality between rule domains and procedures. The process of rule automatization gradually replaces procedural complexity with structural complexity (e.g., Scandura, 1981).

(2) Rules also include ranges, or structured sets of conditions corresponding to anticipated rule outputs.
From a behavioral point of view, rules correspond as naturally to production systems as they do to individual productions. Production systems, however, consist of simple lists of productions which both generate outputs and control which production is tested next. Rule procedures, on the other hand, reflect the branching structure directly. This enhanced structure of rule domains and procedures is what provides the natural basis for assessing and representing individual knowledge. (For further discussion, see Scandura, 1977.)

![Diagram of Subtraction Rule](image)

**Figure 1.** Subtraction Rule (algorithm) and sample subtraction problems.

More important, a long series of studies starting back in the 1960's has shown that the rule representation has direct behavioral implications (e.g., Scandura, 1968, 1969, 1970). If a rule is prototypic of the way (some) individuals solve given classes of problems, then the rule structure provides a natural way to represent individual
knowledge of these individuals. For example, consider the subtraction procedure in the figure shown above.

The obvious thought in this regard is that individual knowledge might be represented by including or not including various components of the prototype. But there is more to it than that. Studies have demonstrated a very direct relationship between individual problem solving behavior and the representation of individual knowledge. In particular, it is possible to operationally define individual knowledge in terms of highly efficient test procedures which derive directly from the rule representation (e.g., Scandura, 1977). There is, of course, a formal equivalent with respect to production systems but whatever that equivalent is, it certainly is not intuitive. Undoubtedly, the strongest argument in favor of the rule representation (as mentioned earlier) is that rules have been favored even by production system enthusiasts in the case of individual knowledge representation.

Level of representation also is important in problem solving research (e.g., Scandura, 1977). Specifically, prototypes used to assess individual knowledge must be compatible both with (a) the sophistication of the target population and (b) the degree of behavioral detail one wants to account for. Put differently, rules must be represented in sufficient detail that the rule components are behaviorally atomic both with respect to minimal capabilities of problem solvers whose behavior is to be explained or predicted and the precision with which that behavior is to be predicted.

Rules, including domains, procedures and ranges, can naturally be represented at arbitrary levels of refinement by simply replacing atomic components with more detailed rules. More important, whatever the level of representation, rules naturally lend themselves to individual knowledge assessment.

Allowing arbitrary refinement in the case of production systems would only make matters worse in this regard. Even simple production systems do not appear to lend themselves well to individual knowledge representation. Hierarchies of production systems would be even less natural.

To summarize with respect to knowledge representation: The basic issue is whether general system considerations (e.g., the ability to handle individual differences, learning, etc.) should be primary as in structural learning theories, or whether personal preference should predominate.
ANALYSIS OF PROBLEM DOMAINS

A second major issue (mentioned above) is how to identify the competence associated with given problem domains. It has become clear that the identification of content-specific knowledge requires input from a domain expert. Many also believe that the services of so-called “knowledge engineers” is crucial (Kinnucan, 1984). Their job is to help domain experts externalize what they know. In this view, the process of knowledge representation is largely an intuitive process.

In contrast, prototypic knowledge in structural learning theories may be identified via Structural Analysis (e.g., Ehenpreis & Scandura, 1974; Scandura, Durnin & Wulfeck, 1974; Scandura 1977, 1982, 1984a, 1984b). Structural Analysis is a systematic (largely algorithmic) process which makes it possible for domain experts to represent underlying knowledge without the services of a knowledge engineer. Put differently, Structural Analysis is a systematic and formalized representation of what knowledge engineers do.

The basic issue, of course, is whether Structural Analysis, or any other method of analysis for that matter, will ever be sufficiently robust — whether there are things that will always remain the province of the knowledge engineer.

“EXTRA DOMAIN” KNOWLEDGE AND UNIVERSALS

Let me turn finally to the issue of “extra domain” knowledge — to influences on problem solving behavior that apparently are external to content specific knowledge. Extra domain knowledge may include logical inference, meaning, induction, problem interpretation, problem solving strategies, conceptual foundations, tacit knowledge or whatever.

How to handle this extra domain knowledge? Some as in the case of expert systems take the position that such things as logical inferencing and induction are best conceptualized in terms of system (or human) universals — in that context, they are often labeled “inference engines.” In problem solving, more generally speaking, these universals are assumed to work in conjunction with domain specific knowledge. Others have made similar universal assumptions regarding such things as generalized “learning to learn” (e.g., McKeachie, 1984), study skills and descriptions of knowledge versus knowledge itself (e.g., Anderson, 1982).
In the Structural Learning Theory, learning strategies, meaning, and induction, as well as all other higher-order capabilities, are treated just as other rules of knowledge (e.g., Scandura, 1973). That is, they are prototypes against which individual knowledge is measured. The only difference in the case of higher-order rules is that they operate on and/or generate lower-order (content-specific) rules. Obviously, viewing higher-order rules as modifiable prototypes gives more flexibility and precision in behavioral prediction. One does not have to assume, for example, that “A or B” and “not A” will ALWAYS lead one to conclude “B” — or that being able to ride a bicycle NECESSARILY means that the person can describe how to do so. Such universals may provide useful bases for Artificial Intelligence or as prototypes. But, they surely are not universally available to humans in all situations.

In structural learning theories only the “goal-switching” control mechanism, along with processing capacity and speed parameters, are considered universal. This division of universals and specific knowledge, has been shown to provide a very natural and modular basis for explaining problem solving behavior, as well as learning, motivation, meaning, memory, automatization and the like (e.g., Scandura 1971, 1973, 1977, 1981).

To summarize, problem solving behavior may be explained in terms of domain specific knowledge supplemented with universal extra-domain knowledge. Or, such (higher-order) knowledge may be considered of the same genre as other knowledge — as prototypes against which individual knowledge is measured. In the latter case, human problem solving (as well as learning, etc.) is explained in terms of a simple, very basic universal control mechanism together with lower and higher order rules associated with (i.e., derivable from) the problem domain.

Each of these views has its advantages and disadvantages. Representing extra-domain knowledge in terms of distinctive, universal inference-engines, for example, may appear to simplify the process of building artificial intelligence systems. Such universals as general heuristics, study skills, and the like also may serve a useful purpose in guiding classroom learning and/or instruction where attention to individual variation is not feasible. However, such systems lack the flexibility needed to deal effectively with individual variation (e.g., in inferencing). Moreover, it is far from clear that representing such universals differently from specific knowledge rules is desirable. At the very least parsimony would seem to suggest otherwise.
There is another facet of the "extra-domain" issue which is not so transparent since it has deeper philosophical roots. Some cognitive psychologists, including some of the contributors to this issue, take the position, explicitly or implicitly, that they are seeking the real truth, that one day we will actually know how people really solve problems. It is just a matter of pinning down the domain specific knowledge and identifying all the extra-domain knowledge involved.

Alternatively, proponents of this view believe no matter how thoroughly a problem domain is analyzed, that problem solving is always subject to outside, or extra-domain, perturbations which must be identified.

The opposite view is that explanation is relative and not absolute. All knowledge is content (i.e., domain) specific, lower and higher-order knowledge alike. The problem domain itself, provides a sufficient basis for identifying all of the relevant knowledge. Using extra-domain knowledge from other sources is at best a convenience which could be eliminated by more complete analysis. With Structural Analysis, for example, all higher, as well as lower, order rules are identified by direct or indirect reference to the given problem domain. The issue of "extra- (or outside the) domain" knowledge never comes up.

Furthermore, it can be shown that beyond a certain point (e.g., Scandura, 1977), further analysis is futile, whether internal or external. Differential predictions resulting from such (further) analysis can not be distinguished within the given problem solving domain.

Consider, for example, a simple problem domain consisting of column subtraction problems in arithmetic. In this case, a simple column subtraction rule would provide a fully-adequate account of performance on that problem domain. An analysis which was extended to include the meaning of the process, say, or means for deriving the subtraction rule, would add nothing in so far as the original problem domain is concerned.

Of course, if the problem domain were enlarged to include manipulations of concrete objects, involving "take-away" say, then the extra-domain knowledge could be crucial. The important point to remember is that the behaviorally relevant parts of such extra-domain knowledge can be derived from the (enlarged) domain itself. Whatever knowledge cannot be so derived will be behaviorally INDISTINGUISHABLE within the problem domain.
SYSTEM ISSUES IN PROBLEM

Another important characteristic of the relativistic view is that rule sets associated with previously analyzed problem domains always provide a useful starting point for analyzing more encompassing domains (e.g., Scandura, 1977, Chapter 2). Progress is strictly cumulative as might be desired in any science.

I could say more about meaning, meta-cognition and understanding but space does not allow. (For this purpose, I would refer you to Scandura, 1970, 1973, 1977, 1980, 1981). The central point I would make here is that meaning and meta-cognition, for instance, are relative notions. Thus, for example, various kinds of manipulations of concrete objects might be viewed as rules in their own right — or, they might be viewed as meanings for other rules, say a rule for performing column subtraction. Thus, a concrete object rule may actually BE meaning in the sense that it is related to another (in this case more syntactic) rule.

What is important in structural learning theories is not only whether something is meaningful, but the nature of the relationship between what constitutes the (syntactic) entity and what constitutes the meaning. That relationship might vary from a simple degenerate higher-order rule, or association, relating the two to a highly general and powerful higher-order rule which operates on any number of meanings and generates their syntactic counterparts. In a similar vein, meta-cognitive processes refer to higher-order rules representing relationships between rule knowledge and descriptions of that knowledge.

CONCLUSIONS

In conclusion, let me briefly summarize the issues.

(1) Do system requirements — such things as provision for assessing individual knowledge, learning, level of representation and identifiability of knowledge — properly impose constraints on the form in which knowledge is represented? I have suggested that they do.

(2) Is the process of knowledge analysis necessarily a human endeavor, or can the process be largely automated as I have proposed? We currently have underway a computer implementation of Structural Analysis which should soon provide a definitive answer to this question.
(3) What is the role of "extra-domain" knowledge? Is knowledge best viewed as relativistic — that is, is all relevant knowledge derivable from given problem domains? Or, is some of the requisite knowledge (i.e., extra-domain knowledge) best viewed as universal, even if tied to particular phenomena such as learning, inference, meaning, understanding, induction, or interpretation (cf. Anderson, 1982, compilation).

Finally, let me call your attention to a common thread underlying each of these issues, and others. Should various aspects of problem solving be studied as isolated phenomena, connected largely by human intuition? Or, is it possible, today, to view problem solving, nay complex human behavior generally, as a rigorously defined, yet unified theoretical system? My own views on the subject must be clear.

References


Scandura, J. M. New directions for theory and research on rule learning: II.
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.