The Effect of Multiple Knowledge Sources on Learning and Teaching

Rogers Hall
Etienne Wenger
Dennis Kibler
Patrick Langley

85-11

Technical Report 85-11

January 22, 1985

Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717

The Effect of Multiple Knowledge Sources on Learning and Teaching†

Abstract

Current paradigms for machine-based learning and teaching tend to perform their task in isolation from a rich context of existing knowledge. In contrast, the research project presented here takes the view that bringing multiple sources of knowledge to bear is of central importance to learning in complex domains. As a consequence teaching must both take advantage of and beware of interactions between new and existing knowledge. The central process which connects learning to its context is reasoning by analogy, a primary concern of this research. In teaching, the connection is provided by the explicit use of a learning model to reason about the choice of teaching actions. In this learning paradigm, new concepts are incrementally refined and integrated into a body of expertise, rather than being evaluated against a static notion of correctness. The domain chosen for this experimentation is that of learning to solve "algebra story problems." A model of acquiring problem solving skills in this domain is described, including: representational structures for background knowledge, a problem solving architecture, learning mechanisms, and the role of analogies in applying existing problem solving abilities to novel problems. Examples of learning are given for representative instances of algebra story problems. After relating our views to the psychological literature, we outline the design of a teaching system. Finally, we insist on the interdependence of learning and teaching and on the synergistic effects of conducting both research efforts in parallel.

† This report is the body of a proposal submitted to the National Science Foundation and to the Office of Naval Research.

The Effect of Multiple Knowledge Sources on Learning and Teaching

I. Introduction

Research in expert systems has shown that knowledge, not search, is the key to expertise. Currently most research in learning and teaching involves a single, narrow arena of expertise. However, neither learning nor teaching takes place in a vacuum. We plan to develop computational models of how learning and teaching interact with existing knowledge. To this end, we plan to construct both a learning system and a teaching system which take multiple knowledge sources into account. By incorporating both teaching and learning components in a single experimental system, we can test a variety of hypotheses concerning underlying representations and processes which support adaptive problem solving behavior. These experiments would be impossible without both components.

Our primary interest is the acquisition of problem solving expertise. As we discuss in more detail later, we have chosen to focus on the specific task domain of solving "algebra story problems." Given their unique educational status, these kinds of problems are particularily appropriate for computational investigation of learning and teaching. On the one hand, these problems are salient within a traditional academic curriculum in that they purport to connect quantitative problem solving skills (e.g., solving for unknowns with simultaneous linear equations) with "real-world" situations which students can be expected to face in adult life (e.g., choosing among different investments or scheduling activities in a work environment). On the other hand, there appears to be consistent anecdotal evidence that such problem solving skills, embodied in instruction on "applied problems" consisting of sets of algebra story problems and techniques for their solution (Kolman and Schapiro, 1981), are poorly taught if taught at all in a classroom setting. Our experiences in mathematical instruction as well as in discussions with secondary school educators confirm Simon's (1980) lament that "textbooks are much more explicit in enunciating the laws of mathematics or of nature than in saying anything about when these laws may be useful in solving problems (p. 92)." Hence, although teaching the problem solving skills necessary in solving algebra story problems is an important goal in most educational curricula, effectively teaching such skills appears to be done haphazardly. If one's purpose is to understand such problem solving skills so that they might be more effectively taught, it is necessary to develop a relatively specific, empirically verifiable model of learning in this domain. Hence, our research has a practical as well as a theoretical orientation.

We intend to show that problem solving begins with problem understanding. We view problem understanding as the correct application of previous knowledge to achieve a unified representation of some problem. As we will discuss, the use of previous knowledge in new situations is directed by a process of analogical reasoning. In the context of multiple knowledge sources, learning is primarily a process of modifying (extending, reorganizing, correcting) background knowledge sources for the purpose of problem solving. Unlike typical machine learning research, we view learning not merely as discrimination or generalization, but rather as concept evolution. In this view, teaching strives to provide support to this evolution by diagnosing the causes of errors and by providing appropriate instruction in terms of the multiple knowledge sources involved.

In the second section we introduce the domain of solving algebra story problems and explain why we have chosen this domain to test our ideas. We also discuss the representation of problems and background knowledge representation, and we propose a candidate problem solver, analogical processes, and learning mechanisms. To further motivate the proposed computational model of learning, we give some examples of its intended behavior. We conclude this section with a discussion of the psychological plausibility of the proposed computational model.

In the third section, we discuss the teaching module. Here we describe the representation of the student knowledge, the formation of a student model, expected errors in student behavior, and corrective actions that the teacher can take. We also consider the relevant literature in Intelligent Teaching Systems (ITS). Finally we discuss the value of interactions between the learning and teaching systems.

II. Learning with Multiple Knowledge Sources

The particular problem domain chosen for this experimental paradigm is that of learning to solve "word" or "story" problems typical of algebra instruction at the elementary and high school level. For example, consider the following "triangle" problem, taken from Mayer (1981):

> *Jerry walks 1 block east along a vacant lot and then 2 blocks north to a friend's house. Phil starts at the same point and walks diagonally through the vacant lot coming out at the same point as Jerry. If Jerry walked 217 feet east and 400 feet north, how far did Phil walk?*

This is called a "triangle" problem since its solution rests on relating the problem to some basic facts about triangles. Other types of word problems that high school students learn to solve, as reported by Hinsley, Hayes and Simon (1977), focus on distant-rate-time, average, ratio, conversion, area, max-min, mixture, simple probability, simple physics, progressions, navigation, number, work, and exponentials. A later study by Mayer (1981) refined this clustering of problem types into approximately 50 distinct problem templates organized as a simple classification hierarchy. Although the intent of these studies was largely to document classification schema which students reliably use to interpret and then solve algebra word problems, the identification and description of these problem schema provides a rare opportunity for choosing a task domain for which there exists a substantial body of descriptive and experimental literature based on human performance.

There are a number of reasons for choosing the domain of algebra story problems:

- the domain is reactive in that the learning component can check the validity of the equations it generates by solving the equations and checking the answers.
- solving word problems involves multiple sources of knowledge including both specific problem templates and supporting background knowledge (i.e., knowledge of geographic orientation would be essential in recognizing a right triangle).
- the particular background or common sense knowledge sources required are numerous (we estimate 20 to 50 sources) but constrained, given the finite set of objects and relationships included in the story problems.
- the domain provides a rich experimental medium intermediate between mathematical and common–sense knowledge reasoning tasks.

- examples of human performance in this domain have been and continue to be the subject of empirical investigation in educational and cognitive psychology, allowing comparison between computational and behavioral approaches to teaching and learning problem solving abilities.

Problem Representation

We will assume that the story problem is presented to the learning system as a relatively flat set of propositions without an explicit representation of structural relations among propositions. For the triangle story mentioned above this representation might be:

motion-event(e1)	*object(e1,Jerry)*	*distance(e1,1 block)*	*direction(e1,east)*
motion-event(e2)	*object(e2,Jerry)*	*distance(e2,2 block)*	*direction(e2,north)*
motion-event(e3)	*object(e3,Phil)*	*beginning(e3,end(e2))*	*end(e3,beginning(e1))*
		distance(e1,217 feet)	*distance(e2,400 feet).*

with the goal of finding *distance(e3,Answer)*. This representation is relatively complete with respect to the literal problem statement and includes information which is extraneous to the desired solution (e.g., that Jerry walks three blocks) but also excludes information which must be inferred if a solution will be found (e.g., that a right triangle has been described). Representations of similar problems described in the psychological literature include inferences from the data and abstractions of important details without describing how such processes take place. In the context of this proposal, such an augmented propositional description represents an intermediate level between our input propositions and more structured forms of representation from which equations utilized in the problem solution can be derived. A test of the intermediate nature of these inferred descriptions can be made by considering slight changes to the problem query. For representations reported in the literature, minor changes in the stated problem require substantial changes to the problem representation.

As will be discussed more fully in describing our candidate problem solving architecture, generation of a problem description which adds appropriate structural inferences and suppresses irrelevant problem details is a central aspect of effective problem solving. This is precisely the sense in which we asserted previously that problem understanding constitutes the beginning phase of skilled problem solving activity by "connecting" a newly encountered problem with existing background knowledge. It is within this context of generating an increasingly coherent problem representation that multiple background knowledge sources play a crucial role.

Background Knowledge Representation

In this section we describe how multiple knowledge sources can provide raw material for use in problem solving and learning as well as in the teaching process. Recent efforts point to the importance of the integration of multiple viewpoints in knowledge representation both in the learning and the ICAI literatures. Burstein (1983) uses multiple analogies to learn concepts in BASIC programming, calling on different analogical sources for different aspects of a developing concept. Rumelhart and Norman (1981) argue that multiple analogical models are necessary to develop and impart a proper understanding of a word processor and its use. They emphasize the need for the teacher to explicitly guide the formation of these models, observing that students will often spontaneously construct their own analogical models and that these models are often deeply erroneous. Brown and Burton (1975) attribute much of SOPHIE's success to its interacting multiple knowledge representations. Finally, and perhaps most definitively, Stevens and Collins (1982), after years of experimentation with their SCHOLAR and later WHY tutors, state as a conclusion that "the step from a unitary to a multi-dimensional representation for instructional knowledge is a necessary one".

While we are strongly committed to the necessity of multiple background knowledge sources interacting in the framing of a problem for solution, we have no particularily strong committment to specific representational techniques for varied knowledge sources. In the following paragraphs we describe *what* types of knowledge sources we plan to include based on an analysis of sample problems found in Mayer (1981), and then briefly consider *how* these multiple knowledge sources might be represented. As will be discussed more fully in the context of the candidate problem solving architecture, specific representational details across varied knowledge sources will be "hidden" by treating each source as a relatively opaque computational object accessible through a uniform propositional interface language.

1. **Methods.** Solution methods will be stored as "schemata", a term used by Mayer, Larkin and Kadane (1984) to denote the generic methods related to problem types in their taxonomy. In our view, the schema includes a cue which is expressed in terms of the features and relations of the problem description (e.g. a motion event where the rate is to be computed) and a solution method which suggests applicable equations.
2. **Events.** Events are central to the description of algebra story problems. Most problems require extraction of quantitative constraints from a single event or multiple interacting events. For the representation of primitive events, we will use frames: each generic event is represented as a set of slots for its different features, and a particular event is a (possibly partial) instantiation of these slots with values. Frames have a long history as primitive knowledge structures in A.I. They were heralded as a general knowledge representation scheme by Minsky (1975) and have since become ubiquitous as organizational structures in a variety of knowledge representation approaches (e.g., Schank's (1975) conceptual primitives). Examples of primitive events relevant to knowledge of algebra story problems would be motion or work events.
3. **Objects.** Physical objects will be organized in a semantic net of frames. This semantic net will be simple since there is relatively little inference needed at the object level in algebra story problems. This is in contrast to the sensitive treatment of inheritance relations advocated by Brachman, Fikes and Levesque (1983). Some object level inferences will be necessary for evaluation of candidate solutions (e.g. a car cannot travel at 200 mph), and the network will provide a conceptual hierarchy for generalization of slot values necessary for concept evolution and elaboration of analogies.
4. **Space.** The representation of spatial knowledge remains an open question in AI. Fortunately, the kind of spatial knowledge required for the understanding and the solution of algebra story problems appears to be relatively simple compared to more general problems of spatial reasoning. Analysis of sample problems presented by Mayer (1981) suggests that we may not need to employ recursively defined hierarchical data

structures such as quadtrees (Samet, 1984) typical in computational approaches to vision. Indeed a rather simple line diagram is sufficient to allow the necessary spatial inferences in sample problems we have studied thus far. Therefore, the representation we have adopted is of the "homomorphic" variety (Hayes, 1974): in a cartesian coordinate system, the relative positions of points and lengths of segments constitute an isomorphic model of the physical relations. Similar diagrams have been used as early as the sixties in proving geometric theorems (Gelernter, 1963) and more recently in Novak's ISAAC physics problem solver (Novak, 1976).

5. **Time.** Representation of temporal relations demonstrates another problem which, when taken in its full generality, remains an open problem in AI. Fortunately, temporal inferences for algebra story problems appear relatively straightforward, underscoring the desirability of this task domain as midpoint between heavily constrained, formal reasoning tasks and the more general difficulties of common–sense reasoning. For our purposes, we should be able to use a version of the interval-based temporal logic proposed by Allen (1983). This logic allows reasoning over relations between intervals which can easily generate the predicates on endpoints that we need.

6. **Facts.** There will be a collection of facts that can be represented propositionally, which we will assume the learner already knows or will be given by the teacher. These facts include specific knowledge of geometry (such knowledge is not equivalent to but may rely upon spatial knowledge), numerical concepts, and monetary knowledge.

7. **Abstract relations.** Finding quantitative constraints within the representation of the problem is the final goal of our problem solving architecture. In their solvable form, these constraints must be expressed as equations. The concept of equality is therefore central to the process of solving algebra story problems. Matz (1982) studies the evolution of the concept of equality between arithmetic and algebra, but she does not propose a representation of the concept. In fact, there is very little in the literature regarding a declarative representation of such abstract concepts. In response we plan to adopt a rule–based representation which should satisfy the needs of our problem solver. Some examples of the we are considering follow:

   a. Two units of measure are compatible iff they are the same or there exists a conversion algorithm between them (i.e., they belong to the same class);

   b. Two things can be compared iff they are expressed in compatible units of measure;

   c. Two quantities can be equated iff they can be compared and their numerical values are the same;

   d. A quantity can be said to be larger than another smaller one iff they can be compared and the larger one can be additively decomposed into one subpart which is equal to the smaller one and another non–null subpart.

8. **Remaining issues.** There are two remaining issues which may prove to be important aspects of background knowledge, but which we are currently deferring. First is the potential necessity of representing and reasoning about the intentions of actors involved in algebra story problems. We are confident that goals (and therefore plans to achieve them) that underly the events of the story problem are important for the characters involved in the story, but do not actively participate in the quantitative derivations of

the solution process. The second issue concerns objects as resources. Time as a relation between events may not be the same as time as a resource which can be consumed. This is especially true in cases where there is competition for resources. Space and money are other examples. Unfortunately, little work has been done on the question of resources in problem solving. An important consideration is the decomposability of resource consuming actions. For instance, the time needed by one person to complete action A and action B is an additive combination of the individual times, but this is not the case if different people perform each action. Morris (1984) describes a formalism which expresses the preconditions of operators in terms of multisets of consumable resources. We are considering using his formalism should the need arise for explicit representation of consumable resources.

A Candidate Problem Solver

In this section we will outline an architecture for the problem solver, the prerequisite representations, and their relationships. To clarify some of the processes and structures we will give snapshots of the intended system's performance.

We regard problem solving in the context of algebra story problems as a particular example of cooperating multiple knowledge sources. Fortunately, relatively well-accepted architectural arrangements already exist for asynchronous management of independent but cooperating knowledge sources via a globally accessible database or context (Erman *et al.*, 1980). For our purposes, background knowledge sources (described in terms of their content in the preceding section) function as relatively autonomous and, by virtue of the opacity of their particular representational medium, anonymous agents which influence each other as well as the course of problem solving by effecting a global database. Within the context of algebra story problems, this database is the current status of a problem description and serves as an interface for interaction among multiple knowledge sources.

Initially, the problem representation is a jumbled set of propositions with very little structure. At each step the problem solver modifies the problem representation, forming a new state description. Each individual knowledge source adds what information it can. This process continues until the goal is reached. The goal is a description which contains a set of equations which correctly solves the initial problem. Thus, the process of problem solving we are proposing is one of constructing an increasingly coherent description of a current problem, using applicable knowledge sources, until sufficient constraints have been collected to derive a plausible solution.

Although the problem solver may utilize a large number of knowledge sources, there appear to be five basic types. While in the previous section we described background knowledge sources primarily in terms of their content, the current description considers knowledge sources in terms of their role in problem solving. This functional view suggests the categories of augmentors, organizers, transformers, assertors, and decomposers. The first three types all change the problem representation by adding propositions. The later two types add equations to the problem representation. Since the goal of the problem solver is to form an appropriate set of equations, and this can only be accomplished by applying the latter two types of agents or knowledge experts, the goal of the first three types is to modify the representation so that the latter types may apply. We now briefly describe each of the five types of knowledge sources.

An augmentor adds information to the problem description by bringing in background knowledge. For example, the information that John flew from New York to Chicago, then on to Los Angeles and finally back to New York implies that John traversed a *triangular* path. That three straight lines connecting head to toe form a triangle is part of background knowledge. As discussed earlier, the various knowledge sources will have representations tuned to their particular form, but the net result of these background knowledge sources is to provide additional propositions important for the problem at hand. Thus augmentation adds pieces of information to the problem description which eventually participate in constraint relations allowing generation of a plausible solution.

An organizer adds information to the problem description by adding structure to the problem. Organizers take pieces of information and put them into a single unit. For example, the fact that John leaves New York and goes to Chicago means that a motion-event has occurred. Since additional information is available about motion-events, a richer and more structured problem description is achieved. Hence, in keeping with our original notion that problem solving efficacy depends in large measure on problem understanding, organizers play a major contributing role in such an understanding by making explicit important constraining relations among constituents of the problem description. The system learns organizers by gradually refining seed concepts provided by the background knowledge.

Transformers take a problem representation and associate with it another problem representation. The process of transforming a problem is a complicated and costly one which, in effect, hides the complexity arising from subprocesses necessary in forming analogies. The subprocesses in analogy formation (recognition, elaboration, evaluation) are treated in more detail in the next section. Transformers relate the current problem to previously solved problems.

Thus far we have considered processing agents whose goal is to alter the problem description so that known methods might be recognized and applied. The next two types of processes involve knowledge sources which generate equations. Hence assertion and decomposition processes correspond most closely to what Mayer *et al.* (1984) intend when discussing problem schemata.

Assertors associate equations with key features of a problem. For example, an assertor would suggest the equation $d = r \times t$ given the cues of a motion-event, known rate and time, and unknown distance. Assertors bring in background equational knowledge that involve a single frame, in this case the motion-event frame.

Decomposers map a problem into subproblems and then relate, via equations, the relationship between the problem and its subproblems. Decomposers are similar to assertors in that they generate equations, but in this case the equations involve more than a single frame. For example in the complex motion problem mentioned when describing augmentation processes earlier, the total westerly distance that John travels can be decomposed as the sum of the distance from New York to Chicago and the distance from Chicago to Los Angeles.

Having described each of five component processes in isolation, it is necessary to give some notion of how these processes will interact through a global problem description to give some movement towards a plausible solution to an algebra story problem at hand. The problem solver schedules activities of the various agents by a competitive process which attempts to do the least work while still progressing towards a goal state. Augmentation and organization are fairly simple and cheap, but do not generate any equations, and therefore may not be useful. Transformation is the most complicated and expensive process agent, but is the activity that most directly relies on and relates to previous problem solving experience. Assertors and decomposers are the only operators that generate equations so must be applied to achieve the goal.

As described, our candidate problem solving architecture is quite general, allowing investigation of a variety of problem solving behaviors with relatively minor alterations of control strategies (i.e., scheduling policies for competing process agents). In a less direct fashion, gradual evolution of conceptual material composing knowledge sources contributing to organization, assertion and decomposition activities should allow a range of problem solving performance from novice to "expert" levels for algebra story problems.

Analogical Transformations in Problem Solving

As we have been discussing our proposed process model for learning problem solving skills in the task domain of solving algebra story problems, we have assumed five basic processes (augmentation, organization, transformation, assertion and decomposition) organized as cooperating process agents within a loose control structure mediated through a global database or blackboard. Augmentation, organization and transformation change the problem description so that assertion and/or decomposition processes are enabled, with a subsequent addition of constraints either in the form of equations or subproblems related in a particular fashion (e.g., total distance as the sum of component distances). As suggested previously, assertion and decomposition processes play the role of "problem schemata" as described in recent psychological studies of problem solving skill with algebra story problems (Mayer *et al.*, 1984).

While this process architecture gives a reasonable beginning for how problem solving might occur with these kinds of problems, the role of transformation processes must be more fully described. The intent of this section is to examine the role of transformation processes in representing the problem in a manner which will lead to a solution. As we have described the transformation process, the central goal is to alter the representation of the current (or target) problem so that it is partially equivalent to a previously experienced (or source) problem for which a solution method (either equations or a suitable decomposition) is known. Stated in the abstract terminology of a search of problem descriptions, the transformation process attempts to change the current problem description so that an existing problem schema (assertion or decomposition process) will be enabled and applied. Thus at an abstract level of description, transformation processes function in a similar fashion to processes of augmentation and organization. However, when viewed more specifically, transformation constitutes a form of analogical reasoning by its explicit goal of viewing a new problem as if it were equivalent to a previously experienced problem. If problem schema are taken to be generalized problem descriptions (e.g., modelled as the left–hand–side of a production) which propose particular methods (either adding equations or related subgoals as the right–hand–side), then an instance of transformation would be the alteration of a target problem description so that is partially matches a schema's generalized problem description. As a result, the methods associated with the matched schema could be applied, possibly leading to a solution directly or decomposing the problem in a fashion which contributes to an eventual solution.

As described, the transformation process is one of viewing an unsolved problem *as if it were* a problem for which a solution strategy were already known. "Viewing as" in this context involves extending information from the solved problem into the description of the new problem, subject to some form of critical evaluation within the confines of the new problem. This is a process of analogical reasoning. By comparison with many traditional views of analogy in the computational literature, this may seem a weak conception of analogical reasoning. However, as suggested by others (e.g., Gentner, 1983), we prefer to think of analogy as one point midway in a continuum between literal similarity and nonsensical (or anomalous) comparison. Given this assumption, reasoning and learning processes engaged while viewing one problem as if it were another more familiar problem are comparable in kind to processes engaged when a new problem is recognized as an instance of a known problem class. Reasoning by analogy, then, may require more effort on the part of the reasoner with regard to confirming extended information, but is not a form of reasoning which can be profitably separated from more mundane forms of problem solving activity (i.e., applying a known problem schema). What remains is an explication of how such analogies are recognized, elaborated and confirmed (Hall, 1984) and how the transformation process functions in learning how to solve algebra story problems.

Recognition of opportunities for transformation

Recognition of a potential analogy within our process model amounts to tentative acceptance of a particular problem schema on the basis of a partial match between the target problem description and the enabling problem description (or source description) contained within the problem schema. In isolation from other processes, it is easy to imagine some form of search among existing problem schemata to find a candidate set of schemata which bear a promising resemblance to the target problem. Adopting this view for a moment, the crucial issue in recognition of an analogy is how this search might be constrained. The psychological literature is relatively mute on this point (with the exception of Holyoak's 1984 discussion of the importance of forming generalized problem schemata), while the computational literature abounds with proposals for constraining such a search. Most promising in these latter contributors is the notion of indexing potential analogs on the basis of abstract relational information (e.g., Carbonell, 1981 and Kolodner, 1983, 1984) so that recognition and retrieval will be based on higher level (and presumably more important) aspects of similarity. Recognition based on similarity between target and source problems for goals, plans or causal structure corresponds closely to Gentner's (1982, 1983) notion of "systematicity" in which effective analogies (for predictive purposes, precisely what we want in solving algebra story problems) reveal similarity in constraining conceptual structure rather than descriptive aspects. In the domain of algebra story problems, a systematic correspondence might be exemplified by a match between a simple motion and a simple work problem which involves some sort of event (e.g., a motion or work event), a rate and a single unknown. While these matching aspects do not directly correspond to goals, plans or causal structure as advocated by Carbonell (1981), they are at a higher level than many other aspects of the participating problem descriptions (e.g., actor's name, starting time, etc.). Such higher level descriptive aspects might be distinguished from their peers by virtue of participating in the method associated with their problem description in the stored problem schema.

Given a form of organization for multiple problem schemata within this task domain which allows indexing on particularily salient aspects of a target (unsolved) problem description, we would like to be able to retrieve some subset of the entire collection of schemata which will contain an effectively analogous problem schema if one exists. Deferring for a moment the question of when search of this memory should be conducted, we still must place the subprocess of recognition within the surrounding model of problem solving. In fact, the process of recognizing a suitably applicable problem schema must be assumed as the central activity in our problem solving model, an activity to which processes of augmentation, organization and transformation are explicitly directed. Hence, we would like to make no special distinction between recognition processes for target problems which directly match an existing (source) problem schema and target problems which partially match an existing schema but require additional effort at elaborating and evaluating (or confirming) the partial match so that problem solving methods contained in the schema might be confidently applied. Thus the difference between "literal" problem understanding and understanding a problem by analogy is a difference in effort rather than a difference in kind.

In summary, we have suggested similar processing mechanisms for routine problem solving and problem solving supported by an analogy. It follows reasonably that search for an analogy is undertaken at precisely the same time that search is initiated for an applicable problem schema. In fact we are arguing that an analogically related schema or a directly applicable schema are retrieved in exactly the same fashion, the latter being distinguished as an "analogy" on the basis of evidencing a less complete partial matching between the target problem description and the schema's problem description (or cue). The recognition process necessary for retrieving directly applicable problem schemata should prove sufficient for retrieval of analogically related problem schemata, providing that relatively weak matches between problem descriptions and schemata cues are allowed. Processes which contribute to the added effort described for confirming extended information from an analogically related problem schema are described next.

Elaboration and evaluation of prospective analogs

Having argued that recognition of analogies is identical to recognition of directly applicable problem schemata except with regard to the effort expended in confirming the match between problem descriptions, we are now in a position of accounting for that effort. Such confirmation is attained by incrementally elaborating the correspondence mapping between source and target problem descriptions. As aspects of the source description (e.g., organizing structural information) are extended to the target description through elaboration, the validity of these extensions must be evaluated within the target problem domain. In the same sense that augmentation and organization processes were described as bringing supporting knowledge sources to bear in understanding a new problem, so too can these processes serve in confirming tentative information extended from source to target problem descriptions.

However, while augmentation and organization have been previously described as *filling in* missing or disorganized aspects of a target problem description, confirmation requires that tentatively extended information be posited as a query to appropriate knowledge sources. For example, when confirming an analogy between a "work together" problem (the target) and an "opposite direction" motion problem (the source), knowledge of equal duration within the motion problem might be extended to the work problem by virtue of being integrally involved in the solution strategy of decomposition. The tentatively held assertion of equal duration in the work problem (extended from the motion problem) must be confirmed on the basis of knowledge which can be generated with augmentation processes. Similar sorts of confirmation could be achieved with organizing processes.

Learning to Solve Algebra Story Problems

A number of different learning mechanisms are supported by the architecture described in preceding sections. Broadly speaking, the system's conceptual knowledge gradually changes as the result of problem solving experience and instruction. Usually some direct instruction will be involved in the extension or correction of many aspects of background knowledge, while learning when to apply background knowledge is the system's responsibility. Hence our computational model of learning to solve algebra story problems involves a variety of forms of learning including learning by being told, learning by taking advice and learning from examples.

The learning mechanisms we propose satisfy a number of constraints. First, they are all incremental. Thus, problem solving expertise is acquired gradually through active experience with a succession of problems and instruction presented by the teacher. Second, and perhaps most importantly, learning is tolerant of errors. Put more strongly, the system can never know that is has a correct conceptualization and consequently the very notion of a correct generalization or concept is misleading in thinking about the proposed system's behavior. Concepts constantly change and evolve according to their problem solving utility. Third, newly gained knowledge is connected to old knowledge. Thus we allow no learning of isolated or disconnected information, a common feature of existing computational learners which we find implausible.

Now we briefly consider four examples of the learning process. These examples show interaction of instruction, background knowledge, and experience. The first two examples are derived from Mayer's (1981) taxonomy. The third and fourth are hypothetical problems designed to mislead an inattentive reader. The first example shows how the system's refines naive concepts into ones more appropriate for problem solving. The second example shows how analogy can suggest transformation of previously learned concepts and schema into new problem solving knowledge. The third example shows how an inappropriate view of the problem, as the result of an insufficiently developed conceptual base, results in an incorrect solution. The last example shows how a combination of problem misrepresentation and an incorrect problem schema leads to missolving the problem.

Before discussing these examples we need to describe the existing state of knowledge of the system. For the first example we suppose that the system has no knowledge of any equations and that the system's naive concept of a motion-event consists of the following frame:

Motion-event:
agent:
vehicle:
to:
from:

Now the system is presented with the following problem.

Example 1: *Bill Less drove from Boston to Cleveland, a distance of 624 miles, in the time of 12 hours. What was his driving speed.*

At this point the system is unable to solve any problems, requiring that the teacher instruct the system to use the equation *distance = rate * time*. At this point, a number of changes take place in the system's knowledge. First, the concept of a motion-event changes to include additional slots for distance, time and rate, since these were used in the problem solution. Moreover, the slots that were used in the solution process are marked as essential. The other slots are noted as potentially inappropriate as they did not play a role in the solution. Second, the system forms a problem schema which associates with the cues, motion-event and goal to 'find rate', the method of using the equation *distance = rate * time*.

After this teaching episode the system's concept of a motion-event would look like:

Motion-event:
agent: (unused 1)
vehicle: (unused 1)
to: (unused 1)
from: (unused 1)
distance: (essential)
rate: (essential)
time: (essential)

Note that the system has demoted the *to* and *from* slots, even though these are important. In fact, even if the system deleted these slots from its representation, the system could still recover by adding them again on the basis of additional experience. As one might easily imagine, with additional examples the system's motion-event concept might evolve to look like the following where all slots are marked "essential":

Motion-event:
distance:
rate:
duration:
start-position:
end-position:
direction:
start-time:
end-time:

The process of adding slots when used and deleting slots that are not used allows the system to avoid distinguishing between correct and errorful concepts. The system drives towards concepts that are useful. The Platonic notion of correct or incorrect concepts is not meaningful to the system. Also slots are not added or deleted merely on the basis of occurrence in the problem description. The inclusion and survival of slots/features depends on their participation in the problem solving process.

Now let us consider a second example which utilizes underlying processes of analogical reasoning to guide learning. We suppose, at this point, that the system knows how to solve simple distance-rate-time problems and it is presented with the following:

Example 2: *A fisherman can catch a fish every 20 minutes. If he spends an 8-hour day fishing, how many fish will he bring home?*

We also suppose that the system contains a naive notion of a work-event which might be represented as:

work-event:
- agent:
- job:
- pay:
- instrument:

The system recognizes that the problem involves a work event, but there is almost no similarity between work-events and motion-events so the system is unable to solve the problem. Now we allow the teacher to suggest that this problem is analogous to the simple distance-rate-time problem. In effect, the teacher has accomplished one of the subtasks of analogy formation, that of recognizing a potential analog. The system now elaborates that analog. In order to do so it must introduce additional features into the notion of work-event, namely a work-rate and a work-time. So again the conceptual language needs to change. Now work-rate and work-time, which are given as one fish per 20 minutes and 8 hours in this problem, can be associated with the speed and time of a distance-rate-time problem. To elaborate the analogy more fully requires that the equation $work = workrate * worktime$ be valid. Once this equation has been confirmed, the problem can be correctly solved. Now a new problem schema can be formed which associates unknown(work) and work-event with the equation $work = workrate * worktime$.

Note that the system had not been told this equation but generated it as the result of an analogy. As presented, recognition of the analogy between simple motion and work problems is provided by the teacher. However, if the system were allowed to attempt all possible analogies, no teacher intervention would be required for this problem. We suspect that such exhaustive search of weak partial matches over existing problem schemata is both cognitively and computationally unreasonable.

The next example shows how incomplete knowledge can lead to an incorrect solution. At this point we refrain from explicitly listing the relevant knowledge structures as they should be clear form the preceding examples.

Example 3. *John hikes for four hours at 3 miles per hour. For the first half of his trip he travels north, then he travels east. At the end, how far away from his initial starting point is he?*

If the system or a student represents the problem as a single motion-event with the goal of finding the unknown distance, then the problem schema developed in the first example will lead to the conclusion that the distance is 12 miles. To correct this, let us suppose that the teacher gives generic advice to read the problem carefully, which is really an instruction to form a better understanding of the problem. In our terms a better understanding is achieved by organizing more of the information within the problem description. In particular, and this may require further instruction, the direction of motion has not been represented. Once the concept of a motion-event is elaborated to include a direction, then the problem can be understood as containing two motion-events. Now the previous problem schema involving a single motion-event would no longer apply. Unless the learner had knowledge of right triangles and of angular relationships among geographic directions, the problem would still be unsolvable, but use of the errorful solution method would no longer occur.

Focusing attention on particularly salient aspects of a problem statement, suggested by the teacher in the previous example, raises important issues regarding differences in problem solving between naive and expert reasoners. It seems likely that substantial differences exist in terms of conceptual structures (Larkin, 1983) and interpretive processes (Clement, 1982) involved in problem understanding which give rise to qualitatively different problem solving behaviors. For our purposes, these differences can be understood in terms of gradually evolving conceptual structures (e.g., introducing a slot for direction into a motion-event frame) which organize problem information with increasing effectiveness and an increasingly important role for analogy as a means of viewing unfamiliar problem situations as variations of familiar problem types.

As an additional strategy for learning *when* to apply particular solution methods, we can envision the learner's generation of specific variations of problem descriptions in a effort to find key problem features for a particular solution method. This is akin to the notion of problem perturbation (Kibler and Porter, 1983b), involving a form of simulation in which values of selected slots of a problem are varied with subsequent problem solving efforts to determine which slot values are criterial to use of a method in question. In the previous example, one might imagine problem variations aimed at determining which problem features appropriately enable attempted application of the Pythagorean theorem (e.g., features which describe a right angle in a triange problem). At present, we imagine that this sort of activity would only be undertaken by relatively advanced problem solvers.

The next example shows an overly general problem schema combining with sparse conceptual knowledge to result in an incorrect solution to a hypothetical problem.

Example 4. *John flies for 4 hours at 100 mph. It costs him* $40 *per hour to fly. How much did the flight cost?*

Assuming the system had an appropriate concept of a motion-event, but had formed an incorrect schema associating a motion-event problem plus an unknown with the use of the distance-rate-time formula, then the system might incorrectly respond with 400. This might happen to a student who did not bother to carefully read the problem and form a representation of all the information within the problem. Certainly one can easily imagine a slight variant of this problem where an answer like 400 was appropriate.

As in the last example, we may imagine that the advice is to reread the problem, i.e. form a more complete problem representation. In particular this problem requires finding an amount of money not a distance. Instructors often try to encourage such a deeper representation by asking students to include units or dimension within the answer.

By modifying the problem schema for distance-rate-time to include units, the learner would not incorrectly apply this schema when the unknown required expression as a monetary amount. By forcing a more complete problem representation, such that the learner records a motion-event, a transaction-event (an exchange of objects) and an unknown cost, schema misapplication can be avoided.

These few examples have shown the nature of conceptual evolution. In particular they illustrate how concepts and problem schema can be learned and unlearned. Naive concepts lead to weak problem representation and understanding, which leads to weak problem solving ability. Larkin (1983) and Clement (1982) note that an important difference between naive and expert problem solvers is that the latter spend more time (proportionally) on problem representation, building more elaborate and integrated representations. This is modelled in our system by the actions of organization processes in concert with frame-like representational structures for primitive concepts. The evolution of such concepts is a necessary step in the development of problem solving skill.

Cognitive Consistency

A sizable body of empirical work has accumulated over the past two decades with respect to problem solving skills involved in solving algebra word and story problems. We intend to shape our computational model of learning and teaching problem solving skills in this task domain so that it is *consistent* with appropriate aspects of this literature, and, where necessary, to supplement this body of empirical evidence with elaborations of existing datasets (e.g., raw "thinking-aloud" protocols for problem solving episodes reported in the literature) or collection of our own evidence regarding the specific form of such problem solving skills and the manner in which they are acquired. In this section, we will discuss existing psychological studies which provide partial answers to important questions regarding problem solving skill within this task domain, and suggest what sorts of further experimentation may be necessary in testing the psychological plausibility of the process model for learning and teaching problem solving skills which we are proposing.

Review of the existing psychological literature

Corresponding to each of the five general process components described in earlier sections of this proposal, relevant psychological research on problem solving skills will be discussed. As will become clear, much of this work is suggestive but hardly conclusive with respect to knowledge structures or processes involved in problem solving generally, and in solving algebra story problems specifically. Furthermore, the psychological literature is sparse regarding how a problem solver might acquire such problem solving skills.

In terms of **augmenting** and **organizing** a problem representation so that it becomes increasingly suitable for solution, we have argued that inferences based on multiple, supporting knowledge sources reflecting "real-world" knowledge play a crucial role. In order for a problem to be solved, it must first be adequately understood. Through processes of augmentation and organization, background knowledge is used to interpret a new problem in a manner consistent with previously experienced problems. The necessity of such inferential activity has long been appreciated in the psychological literature. Paige and Simon (1966), using protocol analysis to investigate problem solving behavior with story problems involving a single unknown, report that "substantive information" (e.g., the monetary value of various coins is positive–valued and fixed for coin type) is used by some subjects in recognizing impossible story problems. Previous computational models (e.g., Bobrow's STUDENT, 1964) viewed such problem solving as a primarily syntactic process of translation and did not predict such behavior. More recent psychological investigation suggests that such inferential activities are important but rely on an effectively usable conceptual vocabulary which many subjects do not possess. Mayer *et al.* (1984), analyzing errors in story recall by college students, found that *relational* elements in the problem statement were most difficult to correctly recall, and that difficulty with such elements resulted in poor problem solving performance. In addition, recall error rates are higher for story propositions irrelevant to the underlying problem type, suggesting that knowledge of problem type is used in organizing aspects of a problem statement during understanding. Evidence from Marshall (1984) shows that roughly one–half of the errors that sixth grade students make on simple word problem are attributable to a lack of problem understanding. These findings suggest that supporting knowledge sources play an important role in representing a problem for solution beyond the obvious application of problem schemata (discussed shortly), and that the integrity of such knowledge sources is problematic even for relatively sophisticated subjects. Hence, we are proposing both that such supporting knowledge sources are used in augmenting a problem representation for solution, and that some aspects of these knowledge sources must be learned if effective problem solving is to be achieved.

With respect to processes which **transform** a target (currently unsolved) problem representation into a source (previously solved) problem representation, a number of psychological studies are of interest. Starting with Dreistadt's (1969) suggestive finding that figural analogies facilitate creative problem solving performance, a variety of researchers have studied the role of transfer between analogous problems. Unfortunately, very little of this work has been done within the task domain of algebra story problems. Studies of relatively unusual problems (e.g., missionaries and cannibals by Reed, Ernst and Banerji, 1974 and a "convergence" problem by Gick and Holyoak, 1980) suggest that naive subjects have considerable difficulty in transferring problem solving strategies between isomorphic versions of these problems. At first glance, such findings might seem to contradict our assertion that analogical reasoning plays a crucial role in learning to solve algebra story problems. However, subsequent experimentation by Gick and Holyoak (1983) aimed at facilitating analogical transfer between related problems supports the importance of learning a generalized "problem schema" in developing problem solving competence. According to Holyoak (1984), such schemata are learned by "eliminative induction" (comparable to "learning from examples" in the computational literature), a process which is indispensable in developing the ability to recognize and apply generalized solution strategies for skilled problem solvers. These schematic strategies are used to reorganize and interpret the problem statement with an eye towards achieving a solution as quickly as possible (termed "solution–focusing by Holyoak).

The studies of transfer between analogously related problems mentioned thus far have been conducted with relatively naive subjects, perhaps accounting for the apparent paucity of analogical reasoning processes contributing to problem solving. According to Gentner (1983), such a deficit could be accounted for by an "analogical shift conjecture" suggesting that novice reasoners draw primarily from literally similar problem solving experiences in solving new problems, since their repertoire of potential problem analogs is limited. With increasing experience, however, their use of analogical reasoning should increase. Clement (1981, 1982), working with skilled problem solvers, reports findings which may corroborate this hypotheses. Describing the results of protocol analysis when experts are presented with challenging physics problems, Clement reports that subjects use chains of interconnected analogies during problem solving, both on the basis of recalling previously experienced problems and by directly generating analogs by altering components of the original problem which are particularily difficult. A process model of subjects' problem solving behavior is described which involves elaborative and evaluative processes quite similar to those proposed here, although issues of recognition and consolidation are not addressed. What is particularily interesting about Clement's work is the manner in which he fits processes of analogical reasoning with a general problem solving context. Analogies are drawn not only in an attempt to identify likely solution strategies, but also in an effort to understand an existing analogy.

As a final example of psychological investigation aimed at understanding the role of analogies in learning problem solving skills for novel problems, Gentner and Gentner (1983) describe a study in which groups of naive subjects were given instruction in electrical circuitry problems on the basis of two underlying metaphors. One group of subjects received a "flowing waters" metaphor in which aspects of circuit problems (i.e., current and resistance) were likened to fluid flow in a closed system (e.g., a battery is like a partially filled water reservoir). A second group of subjects received a "teeming crowds" metaphor involving crowds of mice running through corridors. According to the authors, subjects receiving the aqueous metaphor should have been better able to reason about battery problems than their counterparts receiving the crowd metaphor, who should have excelled on resistance problems. Unfortunately, the empirical results of problem solving performance are not clearly in favor of these expectations, particularily for more difficult circuit problems. However, this study can be criticized for confounding problem solving performance with subjects' background knowledge of both source (i.e., fluid flow) and target (i.e., circuitry) domains as well as presenting problems which may well have exceeded the abilities of their subjects. It would be interesting to repeat this sort of experimentation with genuinely naive subjects and more accessible problems.

The two final process components described in our proposed model, processes which **assert** useful constraining equations and processes which **decompose** a complex problem into related subproblems, have received rich empirical support in the psychological literature on solving algebra word and story problems. Hinsley *et al.* (1977) give clear evidence for the importance of problem schemata reflecting problem type and associated solution methods in a series of experiments designed to demonstrate the existence and use of these schema. These studies reveal that subjects can reliably (across subjects) identify problem types, that such categorization of problems occurs early in understanding the problem (after approximately 18% of the problem has been read), that subjects can accurately predict what sorts of information will come later in the problem before reading further, and that subjects can state problem solving strategies which will prove effective even before reading the entire problem. Thus problem schemata containing useful problem solving methods appear to exist and to be used by experienced subjects in understanding and then solving new algebra story problems. In fact, these authors report further experiments suggesting that subjects attempt to apply problem schemata even in the absence of contextual cues (subjects were presented with semantically nonsensical cover stories), and that irrelevant contextual cues placed within the problem text can cause some subjects to incorrectly categorize problems, resulting in misinterpretation consistent with an inappropriately applied schema.

Also interested in the acquisition and use of problem schemata, Mayer *et al.* (1984) report on a variety of experiments based on errors evident during cued recall of algebra story problems. Frequency of occurrence of particular algebra stories in a large sample of textbooks (1097 problems were collected) correlates positively with accuracy of cued story recall, suggesting that problem types most commonly encountered during subject's educational experiences are precisely the types most likely to be learned as problem schemata and later applied when understanding a new problem. As further evidence for the acquisition of problem schemata, conversion errors (mistakenly categorizing one problem type as another) favor higher frequency problem types from the textbook sample. Hence, in addition to providing evidence for augmentation and organization processes in representing algebra story problems for solution (discussed earlier), these authors provide plausible evidence for the acquisition and use of problem schemata.

In summary, our focus on learned problem schemata as a major contributor in the acquisition of problem solving skill for algebra story problems is consistent with existing psychological evidence. In addition, some empirical results (e.g., Mayer *et al.*, 1984) are suggestive of the importance of acquiring the ability to effectively apply supporting knowledge sources in the organization of a new problem representation. Each of the five process components discussed in our proposed model appears consistent with some aspects of the empirical literature concerning the acquisition and use of problem solving skills in algebra story problems and related task domains. However, as suggested previously, none of these empirical findings are conclusive with respect to knowledge structures and processes involved in acquiring problem solving skill. It is our hope that by the judicious construction of computational models of learning and teaching problem solving skill, we may be able to better understand at a more specific level the sorts of structures and processes which support effective problem solving.

Experimentation

Descriptive studies (e.g., Kilpatrick, 1967; Clement, 1981, 1982; and Larkin, 1983) of problem solving activities in complicated task domains appear to be the norm. We plan to make use of existing datasets (e.g., Kilpatrick, 1967 reports collecting extensive "thinking–aloud" protocols for subjects engaged in solving algebra story problems) as possible, re–analyzing protocols in terms of the process components described in this proposal. As necessary, we have ample access to elementary, secondary, and post–secondary school students as subjects in descriptive or experimental studies. Of particular interest to us is investigation of the facility with which subjects of varying skill levels recognize, elaborate and evaluate analogies when solving algebra story problems. A clearer empirical picture of this problem solving strategy should have important implications for computational approaches to learning and teaching problem solving skills in this and other domains.

Prescriptive experimental studies are more difficult to conduct, but potentially more revealing. Such studies in the domain of problem solving typically involve manipulation of task materials and stimulus setting as independent variables, and various measures of problem solving performance as dependent variables. Methodological problems in past studies of problem solving behavior have been selection of task difficulty (e.g., Reed, Ernst and Banerji, 1974) and control of subject's background knowledge (e.g., Gentner and Gentner, 1983). Algebra story problems in some measure resolve the former difficulty, since most subjects can be assumed to possess sufficient background knowledge for understanding the story line typical of such problems. A focus on learning with naive subjects (e.g., selecting subjects before they have been exposed to ASP's or doing stratified random sampling with college students) may relieve the latter problem with respect to specific quantitative knowledge.

Again, our interest in the acquisition of conceptual and strategic knowledge which facilitates problem solving in this domain is of primary importance. We are hypothesizing that analogical reasoning plays an important role in learning and teaching of specific problem solving skills. Empirical verification of this hypothesis must illuminate processing issues of recognition, elaboration, evaluation and consolidation which compose the constituent activities in transforming a novel problem into another, better understood, problem. At present, we expect to conduct relatively simple experiments in which subjects are presented with a choice of analogical sources of varying appropriateness in solving a novel problem by virtue of having solved each of these sources in the immediate past. Choice of analogical source and performance in subsequent problem solving should be revealing with respect to subprocesses composing the analogical transformation of a novel algebra story problem.

III. Teaching with multiple knowledge sources

Now that we have presented our performance and learning models, and given support for their validity, we can turn to the description of the teaching system that will use these models as a foundation. Our performance model will provide the explicit statement of the expertise to be conveyed. Since the necessity of such an explicit model has long been recognized and since the model has been described in details, we will not say much about it in this section. What is novel and will be emphasized is the use of an explicit learning model to guide the way in which the expertise is conveyed.

A perspective on teaching

To our knowledge, no currently implemented teaching system deals explicitly with the multiple sources of knowledge that students bring to bear while learning. The "overlay" student modelling paradigm (Carr and Goldstein, 1977) used in many systems, rules and malrules (Sleeman, 1983) and the subskill lattice proposed for Debuggy (Burton, 1982) all concentrate on relatively independent skills in some isolated domain. The "genetic graph" (Goldstein, 1982) does include explicit links between pieces of knowledge, but these links are static and do not cut across domains.

Some recent theoretical studies attempt to dynamically relate the acquisition of new knowledge to previous knowledge in the same or in another domain. For example, Matz (1982) studies the transition from arithmetic skills to high-school algebra. To a certain extent, the repair theory (Brown and VanLehn, 1980) and its successor, the step theory (VanLehn, 1983) also accounts for processes that take place when old or incomplete knowledge is faced with new situations, although from the restricted perspective of surface repairs. Most importantly, these theories are oriented toward explaining the emergence of errorful behavior, and do not claim to be general learning theories. Furthermore, no effort has been invested so far to fully evaluate their implications for teaching.

Furthering these recent efforts, we take the position that the student's learning is greatly affected by his previous knowledge, and that teaching must necessarily be performed in this context if it is to successfully foster the integration of new information. Although we will certainly draw on the findings of the aforementioned theories, our own learning system will provide the main theoretical basis for our teaching approach, since it is based on the same views. A similar methodology is being applied by Anderson and his colleagues at CMU who are building intelligent tutors on the basis of their theory of skill acquisition (Anderson, 1984).

Our domain is particularly well suited for an investigation of teaching with multiple knowledge sources for many of the same reasons as it is for learning. In addition, the ease of task generation for specific problem schemata, and the interactive guidance needed for problem solving provide an ideal framework to study the kind of "connected" teaching that we advocate. Because we have access to previous research on the knowledge sources involved in solving story problems, and because these sources are of limited scope and can be well defined, we expect to be able to:

- model the student's knowledge and follow the skill acquisition process in terms of the interactions between knowledge sources

- take advantage of the interactions between knowledge sources to present new information, to guide the student and to form a teaching plan.

A generally accepted framework for intelligent teaching systems was presented by Hartley and Sleeman (1973). It comprises: an *expert* module (knowledge of the domain), a *diagnostic* module (student model) and a means-ends *teaching* module (teaching strategies). Some researchers have also included an explicit representation of a syllabus. We will follow this division in our presentation (in a sightly different order). In this project, we want to view teaching as a series of transformations on the state of knowledge of the student, achieved by an iterative process of diagnostic and tutorial actions. Therefore we will emphasize the complementary functions of the diagnostic and the teaching modules.

The diagnostic module

The diagnostic module is comprised of a data structure, the *student model*, which is executable like most such models in recent teaching systems, and of a *diagnostic process* which updates the student model. We will review them in separate sections for clarity.

The structure of the student model

In the student modelling paradigms mentioned above, the knowledge of the student is represented as a subset of a representation of the domain skills, in some cases augmented to include buggy behaviors. For our student model to effectively reflect the effect of multiple knowledge sources, it must be provided with learning capabilities which we expect the student to have. We want to explore the feasibility of a new paradigm: a student model which is not a subset of an extended syllabus, but an active learning entity by itself. A rudimentary form of such a student model has been proposed by Self (1977) in his interesting concept teaching experiment. In our case, we claim that these learning capabilities can direct the search for a model in a way which is tightly connected to the teaching history as well as offer an experimental testbed for the teaching module.

Because we are independently developing a corresponding learning system, we have a computational model of our current learning theory, and we can incorporate *within* the teaching system a version of the learning system to be used as a student model. Being active and somewhat independent, this model can be triggered by other modules with specific parameters to provide useful information. Now having a completely separate problem solver for the student may not be desirable. The overlay modelling paradigm has the advantage that it avoids duplication of knowledge representation for the expert and the student. Not only is such duplication wasteful in terms of memory space, but it ignores the natural relations that exist between the two structures. To have the best of both worlds, we will establish links between the distinct student model and the expert's knowledge representation. These links will either claim equivalence between some subparts of the knowledge, in which case duplication can be avoided by providing access to the expert knowledge, or report some difference in which case separate representations are at least partially required (e.g. for a naive concept of a motion-event). Although this looks like an extended overlay, it is important to understand that these links are not static pointers between two knowledge structures, but are a dynamic window through which parts of the expert knowledge can be indexed by an active learning model.

Apart from these connections to the expert, the main difference between the internal student model and the external learner being taught (eventually real students) is that the teaching system has direct access to the internal representation of its own student model, but only communicates with the external learner through a (formal) language. The internal student model also contains some ancilliary information concerning the frontier of its current knowledge, its learning history and its preferred learning strategies.

The paradigm of a transparent student model based on a theory of learning with multiple knowledge sources has enormous potential because it links teaching to learning in a very direct way. It is an interesting development of recent ideas by Langley et al. (1984a, 1984b) who use machine learning techniques to infer the student's procedure from his behavior by trying to learn the procedure that would produce such a behavior. However, we go one step further in that we use our learning model to search for the student's state of knowledge, taking, like Langley, advantage of the fact that the learning techniques can both learn and mislearn. Whereas Langley et al. use learning techniques for the purpose of inducing the performance model independently of how it was learned originally, we use our learning model as an operator to traverse the space of possible performance models so that we can take advantage of the teaching history. We will now describe how we manage the enormous search space generated by such a complex operator by placing the search in the context of the teaching sequence.

The diagnostic process

Since we propose to interleave teaching and diagnostic actions, the management of the student model takes place in two phases. First, it must take into account the last teaching action, then it must perceive the actual effect of the teaching action on the external learner. While a teaching action is being performed, the student model can be made to tentatively reflect its expected effect. Since the student model is an active learning model, this can be done by running a simulation. Alternatively, a direct modification by the teaching module is conceivable. In this sense the first use of the student model is to act locally as a perfect student. The diagnostic process can then compare the behavior of the external student to that of its internal version and modify the latter to be a model of the external student. In this framework, not only is it possible to compare the student's performance to the expert's, but also to that of a locally perfect student model. Comparison of the student's behavior with the expert's will lead to the formulation of an overall teaching plan, while comparison with the perfect student's will generally lead to local teaching decisions.

In general, tuning the internal model to the external student is an extremely difficult task requiring very powerful inference methods (Sleeman, 83). This is especially true in a domain like ours where opportunities for direct observation of the student's problem solving behavior are of necessity relatively rare*. We have suggested (Wenger, 1985) that a purely inductive approach to student modelling may be unnecessarily difficult in many cases. Even with very powerful inference capabillities, it may never be possible to model the student with enough precision given the number of extraneous factors that can be relevant. Indeed we suggested that the student modelling process should include interactions with the student, and be interwoven with tutorial interventions, whenever it is the case that the student can articulate his decisions. Interviewing young students about their knowledge of simple algebraic manipulations, Sleeman (1984) reports having found them to be very articulate in describing their own approaches. This indicates that students solving algebra story problems which come later in the curriculum can be expected to answer reasonable questions about their problem solving. More recently, Greeno (1984) is taking an interactive approach in his algebra tutor where he requires the student to

* It may be argued that other domains like programming (in recent tutors for Pascal (Johnson and Soloway, 1984) or Lisp (Farrel et al., 1984)), provide more opportunities for such observations. However, observing the code is more informative only if one just teaches a programming language in isolation. A written program is only the end result of a long reasoning process. If one teaches program design, the difficulty is the same as in our domain.

indicate his understanding of the structure of a problem as well as his relevant background knowledge with a graphical editor. Although his main purpose is to enforce a deeper and clearer perception of the relations underlying the problem statement, the information provided by the student has great diagnostic value.

Therefore we will provide mechanisms to interactively confirm or refine hypotheses about the student's knowledge in addition to (not in replacement of) inductive capabilities. To this end we plan to allow the diagnostic process to generate a limited number of questions which can be inserted into the teaching dialogue in concert with the teaching module. We envision the two modules cooperating to conduct a dialogue with the student, probing his knowledge while providing guidance for problem solving (see next section). The use of such interactions simplifies the diagnostic process in two ways. First the search can concentrate on specific parts of the model in the context of the tutorial dialogue. Secondly the confirmation of the student model can be partial since further refinements may be provided by later questions when required.

We have argued that even with good interactive facilities, the student modelling process will have to apply induction to some extent (Wenger, 1985). There are facts about our diagnostic process which make the search somewhat easier independently of our interactive facilities. First, while it is true that both the Debuggy system (Burton, 1982) and LMS (Sleeman, 1983) had to deal with combinatorial explosion during the diagnostic search even for very simple domains, it should be noted that the search was really taking place *out-of any learning context*. Because our model incrementally takes into account the teaching sequence and the interactions between knowledge sources influencing learning (e.g. wrong transfers in analogies) and problem solving (e.g. wrong cuing for a problem) we expect to be able to guide the search in a sensible way. Indeed, when the student model is first modified to reflect a teaching actions, points that are prone to mislearning are already flagged as such and can be considered first, should the student manifest difficulties. Such dynamic priming of potential errors is impossible without a learning model.

Secondly, we plan to conduct a detailed analysis of likely sources of errors associated with individual knowledge sources. This effort will be greatly facilitated by large collections of data provided by other studies, in particular (Marshall, 1984). A priori knowledge of likely error sources is essential for the search involved in the modelling task to be efficient. So far the sources of errors we have identified fall into four general classes for a given piece of background knowledge:

- it may be missing;
- the opportunity to use it may be overlooked;
- it may be incorrect or incomplete;
- it may be misapplied.

For the equation applicable to the triangle problem that we have been considering, these classes of errors could manifest in the following ways for a student:

- he does not know the equation $hyp^2 = leg1^2 + leg2^2$;
- he does not notice that the legs of the triangle are perpendicular;
- his equation reads $hyp^2 = leg1^2 \times leg2^2$;

- he applies this equation to a problem where the two legs are not perpendicular.

It is interesting to note that, with the exception of mere slips of performance, all the errors we have observed up to now can be traced to one of these classes of errors for one or more knowledge sources. This is made possible because we view the process of understanding the problem, that is, moving from the given problem representation to the equations, as a series of incremental contributions by different knowledge sources. Apart from its value as a problem solving and learning model, this view has profound implications for the kind of remedial actions which can be taken by the teaching module, correcting errors at the knowledge source level rather than at the problem level. We feel that this is a very promising direction.

The teaching module

At the outset of the project, we will not be concerned with the questions of the output format in which the information should be presented to the student, although these are important issues. Working with a formal language will free us from the problems of natural language processing or graphic generation. Rather we will deal with the basic issue of determining which pieces of information are relevant at a given time and in which context they can be related to other pieces of knowledge.

As mentioned in the preceding section, much of the tutoring will be done by means of a dialogue conducted by a cooperation of the diagnostic module and the teaching module. In the first stage, this dialogue will take place in the formal language mentioned above (which can later be translated into a subset of English for human students). Although the state of the art in language processing does not support free dialogues, research in teaching systems has been met with a fair amount of success in interactive dialogues with students in limited domains (see Sophie's dialogue facilities (Burton and Brown, 1977) and Ace, a program that analyzes students' explanations of their reasoning (Sleeman and Hendley, 1982)). The question-answer approach to guidance for problem solving has been successfully adopted by a group at UCI's Educational Technology Center (Bork, 1981) (Trowbridge and Chiocarello, 1985) in their highly regarded CAI projects for physics problems, and we will benefit from their experience.

With the information provided by the performance and the learning models as well as the information it receives from the diagnostic module, the teaching module will fulfill four functions:

- establish a teaching plan;
- present general knowledge that the student needs or which is on the plan;
- generate new problems on the basis of the teaching plan, the current student model and the problem templates in its curriculum;
- supervise the process of solving one problem by the kind of interactive guidance described above. This function will build on some extensive research on coaching strategies done by Brown and Burton (1982), Goldstein and Carr (1977), and recently by Farrell et al. (1984).

In selecting tutorial actions, the teaching module will follow principles that parallel those guiding the learning system, i.e. it will generate a teaching sequence which presents information to the learner

- in an incremental way: only moderate changes of the student's model are required;
- in a connected way: the new material relates to correct knowledge that the student possesses, and in many cases the connectedness is explicitly used in the teaching process;
- in a reactive way: the student will be required to use his new knowledge on subsequent problems.

One interesting use of the active student model is in experimenting with teaching decisions on the student model before really acting on them with the external learner. O'Shea (1979) perceived that the fact that his student model was too primitive for that kind of dynamic experimentations was a severe limitation of his self-improving teaching system. It will be very useful to find out how much such experimentation on the part of the system improves the teaching performance. If the improvement is substantial, it could be hypothesized that the teacher's ability to momentarily become a virtual student and receive his or her own teaching from a student's point of view is a critical teaching skill.

Scope, syllabus and expertise

Expertise in solving story problems, like most forms of expertise, covers a range of abilities from generally applicable weak methods to highly specific knowledge. Without general strategies and understanding, the expertise is limited to a small set of well-structured problems, but fails in all other cases, even when a familiar problem is presented in an unusual way. Without highly specialized knowledge, the novice does not have a good framework to organize the problem in a way that is readily amenable to a solution (Larkin, 1983). Obviously, there is a continuum of subskills between the two extremes. Our knowledge-based approach to learning and teaching has the uncommon advantage of covering the entire spectrum in a unified fashion which accounts for this continuum. Recall that we view the process of understanding a particular problem as an incremental structuring of the problem representation by knowledge sources implemented as independent agents. These knowledge sources include both very specific knowledge, like schemata for different problem types, and general background knowledge, like time, space, etc.

The skills at the domain-specific end of the spectrum are performance-oriented and are strongly connected to the acquisition of problem schemata, of well-tuned conceptualizations and of matching skills to retrieve relevant templates early in the solving process. These skills can be fostered by judicious teaching plans taking into account the underlying connections between example problems and by appropriate hints from the teacher.

The weak methods include problem-solving strategies and the ability to understand the situation described by the problem. For the purpose of this project, we want to measure this latter ability by the quality of the problem representation that the student builds before the problem is related to previously encountered problem types. In the context of analogical reasoning, we define this as the amount of structural information that is used in drawing analogies. We claim that this deeper understanding of the problem hinges on correct application of background knowledge to structuring and augmenting the problem description. Because our diagnostic and remedial actions are precisely oriented toward background knowledge and its applicability, they will help the student develop the ability to deal with new problems. To enforce problem understanding, the teaching module can introduce in the given statement some pieces of information whose relevance can only be determined after a relatively complete structured representation of the problem has been reached.

The interaction between the two systems

Although some work with simulated students has been done (Goldstein, 1982) (Self, 1977), our situation is unique in the sense that our learning system is not merely developed as a testbed for a particular teaching system, but as a machine learning project in its own right. By working on both a teaching and a learning system simultaneously we will benefit from a shared exploration of underlying representations and processes. In addition, the interactions taking place in the course of these two parallel investigations will reflect the natural interdependence of theories of learning and teaching. The nature and implications of this interdependence is a major theoretic issue central to our research on teaching.

Furthermore, if care is taken to parameterize the characteristics of both systems, we can link them to explore their respective features by holding one fixed and varying the other. Although there can be doubts concerning the general validity of such "closed loop" testbeds, it should be stressed again that it is not either system's primary purpose and that these experiments will be only used as preliminary tests.

Such closed loop experiments in the first testing stage offer a number of advantages. Different levels of teacher intervention and problem generation can be precisely defined to evaluate different learning algorithms. Different teaching plans can be evaluated by the correctness and completeness of the knowledge structures acquired by the learning system. This ability to "see" the knowledge structures and processes of the student provides us with a first evaluation of the teaching system, without having to deal with the problem of abstracting the actual solution process from indirect evidence such as verbal protocols. However we do intend to test the teaching system with human students.

IV. Conclusions

Taking into account the effect of multiple knowledge sources is a timely step in the current research lines of both machine learning and ITS. The majority of computational approaches to learning reported thus far have investigated isolated task domains in which the learner is implicitly provided with an attentional focus on important aspects of the learning task. In contrast, learning to solve algebra story problems provides a rich, relatively open-ended task environment in which inferences must be drawn from a variety of diverse sources of knowledge. Likewise, compared with traditional research in ITS, our task domain provides a challenging medium for investigating issues of curriculum planning, student modelling and instruction. The central role played by analogical reasoning in bringing multiple knowledge sources to bare in understanding and solving algebra story problems presents novel opportunities for investigation of learning and teaching alike. Many of these opportunities stem from our interest in making learning and teaching "connected" to previous knowledge. Hence our choice of an application domain midway between commonsense reasoning and formal problem-solving allows us to attack challenging problems without having to encode an unreasonably large amount of background knowledge. By combining parallel but independently developed learning and teaching systems into a single experimental setting we are recognizing the duality of theories of learning and teaching in a synergistic fashion. Our system's potential as a testbed for theoretical issues will make it an important contribution for both fields.

Taking a more specific viewpoint, the proposed research is novel in several important ways. First, we are interested in computational models of what we term "conceptual evolution," a form of concept acquisition guided by utility in problem solving performance rather than *a priori* criteria of conceptual correctness. Hence we are proposing a form of learning which is inherently errorful but remarkably robust with respect to a changing external environment. In our view, this form of conceptual adaptation appears much more consonant with human performance and should allow a wide space of potential performance levels for our computational problem solver.

Second, the relatively unrestricted use of analogical reasoning during transformational processes of problem understanding represents an investigation of analogical processes in problem solving which anticipates an apparent trend in computational studies of analogy. Rather than viewing analogy as a special or uncommon process invoked occassionally when more routine problem solving strategies have been exhausted, we argue that viewing one problem situation as if it were another, more familiar problem situation is an ubiquitous component of effective problem solving. In our view, new problem situations must be understood in a connected fashion with existing knowledge sources if problem solving is to be possible. Analogical reasoning provides this sort of connected understanding.

Third, the open–ended nature of expertise in algebra story problems offers a new challenge for a teaching system. Our view of problem understanding as the successive application of background knowledge sources to the current problem provides a performance model which focuses diagnostic and tutorial actions on the concepts and misconceptions underlying the problem solving activity. While this view lifts the teaching beyond the solution of specific problems, we still recognize the pedagogical importance of dealing with the schemata that people do develop and associate with different problem types. Thus our approach to teaching addresses a wide range of problem solving skills essential for expertise in this domain.

Fourth, we propose that the diagnostic and teaching processes be woven into one interactive process consisting of alternating diagnostic and tutorial actions. This requires that both processes cooperate to generate a coherent dialogue, but it allows student modelling to be incremental, and tightly connected to instructional units. Furthermore we suggest that the derivation of the student model be a combination of inductive inferences based on observation of the student's actions and direct interactions with the student.

Finally, the central role played by the learning model in the teaching functions makes an explicit use of the interdependence of teaching and learning. This interdependence is manifest both in the diagnostic function where the learning model is used to connect the diagnostic process to the teaching history and in the formation of teaching actions where it provides guidance by helping predict the effects that can be expected from these actions. We plan to study the nature and implications of this interdependence in the context of teaching systems, claiming that it is one of the most important issues identified by more than a decade of research in ITS.

V. References

Allen J.F. (1983) Maintaining knowledge about temporal intervals. CACM, Vol. 26, Number 11, pp 832-843.

Anderson J.R. (1983) Acquisition of proof skills in geometry. In R.S. Michalski et al. (Eds.) *Machine Learning.* Tioga Publishing Company: Palo Alto, California.

Anderson J.R. (1984) Cognitive psychology and intelligent tutoring. *Cognitive Society Conference*, pp. 37-42.

Bobrow D.G. (1968) Natural language input for a computer problem–solving system. In M. Minsky (Ed.) *Semantic information processing.* The MIT Press: Cambridge, Massachusetts, 135–215.

Bork A. (1981) Computer-based instruction in physics. *Physics Today*, September.

Brachman R.J., Fikes R.E. and Levesque H.J. (1983) Krypton: a functional approach to knowledge representation. *Computer*, October, 67–73.

Brown J.S. and Burton R. (1975) Multiple representations of knowledge for tutorial reasoning. In Bobrow D. and Collins A. (Eds.) *Representation and understanding*, Academic Press.

Brown J.S. and Burton R. (1982) An investigation of computer coaching for informal learning activities. In D. Sleeman and J. S. Brown (Eds.) *Intelligent Tutoring Systems.* Academic Press, London.

Brown J.S. and VanLehn K. (1980) Repair theory: a generative theory of bugs in procedural skill. *Cognitive Science, 4*, 379-427.

Brown R. (1977) Use of analogy to achieve new expertise. AI Lab MIT, AD-AO43809/3.

Burstein M.H. (1983) Concept formation by incremental analogical reasoning and debugging. *Machine Learning Workshop.*

Burton R. (1982) Diagnosing bugs in a simple procedural skill. In D. Sleeman and J. S. Brown (Eds.) *Intelligent Tutoring Systems.* Academic Press, London.

Burton R.R. and Brown J.S. (1977) Semantic grammar: a technique for constructing natural language interfaces to instructional systems. *Bolt, Beranek and Newman Report 3587 (ICAI Report 5).*

Carbonell J.G. (1981) Invariance hierarchies in metaphor interpretation. *Proceedings of the third annual Cognitive Science Society*, 292–295.

Carbonell J.G. (1983a) Learning by analogy: formulating and generalizing plans from past experience. In R. Michalski et al. (Eds.) *Machine Learning.* Tioga Publishing Company: Palo Alto, CA.

Carbonell J.G. (1983b) Derivational analogy in problem solving and knowledge acquisition. *Machine Learning Workshop.*

Carr, B. and Goldstein, I. (1977) Overlays: a theory of modelling for Computer Aided Instruction. *MIT AI Memo 406.*

Clement J. (1981) Analogy generation in scientific problem solving. *Proceedings of the third meeting of the Cognitive Science Society*, 137–140.

Clement J. (1982) Analogical reasoning patterns in expert problem solving. *Proceedings of the 4th Annual Conference of the Cognitive Science Society.*

Dreistadt R. (1969) The use of analogies and incubation in obtaining insights in creative problem solving. *The Journal of Psychology 71*, 159–175.

Erman L.D., Hayes–Roth F., Lesser V.R. and Reddy D.R. (1980) The HEARSAY–II speech understanding system: integrating knowledge to resolve uncertainty. *Computing Surveys 12*(2), 213–253.

Farrell R.J., Anderson J.R. and Reiser B.J. (1984) An interactive computer-based tutor for LISP. *Proceedings of AAAI*, pp. 106-109.

Gelernter H. (1963) Realization of a geometry–theorem proving machine. In E. Feigenbaum and J. Feldman (Eds.) *Computers and Thought.* McGraw Hill: New York, 134–152.

Gentner D. (1982) Are scientific analogies metaphors? In D. Miall (Ed.) *Metaphor: Problems and Perspectives.* Humanities Press: New Jersey.

Gentner D. (1983) Structure–mapping: a theoretical framework for analogy. *Cognitive Science 7*, 155–170.

Gentner D. and Gentner D.R. (1983) Flowing waters or teeming crowds: mental models of electricity. In D. Gentner and A.L. Stevens (Eds.) *Mental Models.* Lawrence Erlbaum Associates: Hillsdale, New Jersey, 99–129.

Gick M.L. and Holyoak K.J. (1980) Analogical problem solving. *Cognitive Psychology 12*, 306-355.

Gick M.L. and Holyoak K.J. (1983) Schema induction and analogical transfer. *Cognitive Psychology 15*, 1-38.

Goldstein I. and Carr B. (1977) The computer as coach: an athletic paradigm for intellectual education. *Proceedings of the ACM Annual Conference*, Seattle, 227-233.

Goldstein I. (1982) The Genetic Graph: a representation for the evolution of procedural knowledge. In D. Sleeman and J. S. Brown (Eds.) *Intelligent Tutoring Systems.* Academic Press, London.

Greeno J. (1984) Personal communication.

Hall R. (1984) Analogical reasoning in artificial intelligence and related disciplines. University of California, Irvine.

Hartley J. and Sleeman D. (1973) Towards intelligent teaching systems. *International Journal of Man-Machine Studies, 5*, 215-236.

Hayes P.J. (1974) Some problems and non-problems in representation theory. *British Computer Society, AI and Simulation of Behavior Group summer conference*, University of Sussex, 63–79.

Hinsley, Hayes and Simon (1977) From words to equations: meaning and representation in algebra word problems. In Carpenter and Just (Eds.) *Cognitive Processes in Comprehension.*

Holyoak K.J. (1984) Analogical thinking and human intelligence. In R.J. Sternberg (Ed.) *Advances in the Psychology of Human Intelligence 2.* LEA: Hillsdale, N.J.

Kibler D. and Porter B. (1983a) Episodic Learning. *Proceedings of the AAAI*,1-196.

Kibler D. and Porter B. (1983b) Perturbation: a means for guiding generalization. *Proceedings of the IJCAI*, 415–418.

Kilpatrick J. (1967) Analyzing the solution of word problems in mathematics: an exploratory study. Dissertation, Stanford University. UMI–68–6442.

Kolman B. and Shapiro A. (1981) *College algebra and trigonometry.* Academic Press: New York.

Kolodner J.L. (1983) Maintaining organization in a dynamic long–term memory. *Cognitive Science 7*, 243–280.

Kolodner J.L. (1984) Towards an understanding of the role of experience in the evolution from novice to expert. GIT–ICS–84/07.

Langley P., Ohlsson S. and Sage S. (1984a) A machine learning approach to student modelling. *CMU-RI-TR-84-7.*

Langley P. and Ohlsson S. (1984b) Automated cognitive modelling. *Proceedings of the AAAI Conference.*

Larkin J.H. (1983) The role of problem representation in physics. In D. Gentner and A. Stephens (Eds.) *Mental Models.* LEA: Hillsdale, N.J.

Marshall S. (1984) The consistency of students errors in mathematical problem solving. Psychology Department, University of California, Santa Barbara.

Matz M. (1982) Towards a process for high school algebra errors. In D. Sleeman and J. S. Brown (Eds.) *Intelligent Tutoring Systems.* Academic Press, London.

Mayer R.E. (1981) Frequency norms and structurral analysis of algebra story problems into families, categories, and templates. *Instructional Science 10*, 135–175.

Mayer R., Larkin J. and Kadane J. (1984) A cognitive analysis of mathematical problem–solving ability. In R. Sternberg (Ed.) *Advances in the Psychology of Human Intelligence 2.*

Minsky M. (1975) A framework for representing knowledge. In P. Winston (Ed.) *The psychology of computer vision.* McGraw Hill: New York, 211–277.

Morris P. (1984) A resource oriented fomalism for plan generation. Dissertation, University of California, Irvine.

Novak G.S. (1976) Computer understanding of physics problems stated in natural language. Dissertation, University of Texas, Austin.

O'Shea T. (1979) *Self-improving teaching systems.* Birkhauser: Basel.

Paige J.M. and Simon H.A. (1966) Cognitive processes in solving algebra word problems. In B. Kleinmuntz (Ed.) *Problem solving: research, method, and theory.* Wiley: New York.

Porter B. and Kibler D. (1984) Learning operator models. *Proceedings of the AAAI.*

Reed S.K., Ernst G.W. and Banerji R. (1974) The role of analogy in transfer between similar problem states. *Cognitive Psychology 6*, 436–450.

Rumelhart D.E. and Norman D.A. (1981) Analogical processes in learning. In J.R. Anderson (Ed.) *Cognitive skills and their acquisition.* Lawrence Erlbaum Associates: Hillsdale, New Jersey, 335–359.

Samet H. (1984) The quadtree and related hierarchical data structures. *ACM Computing Surveys*, Vol. 16, Number 2, pp. 187-260.

Schank R.C. (1975) *Conceptual information processing*, North Holland, Amsterdam.

Self J. (1977) Concept teaching *AI Journal, 9*(2), pp. 197-221.

Simon H.A. (1980) Problem solving and education. In D.T. Tuma and F. Reif (Eds.) *Problem solving and education: issues in teaching and research.* Lawrence Erlbaum Associates: Hillsdale, New Jersey, 81–96.

Sleeman D. and Hendley R.J. (1982) ACE: a system that analyses complex explanations. In D. Sleeman and J. S. Brown (Eds.) *Intelligent Tutoring Systems.* Academic Press, London.

Sleeman D. (1983) Inferring Student Model for Intelligent Computer-Aided Instruction. In R. Michalski et al. (Eds.) *Machine Learning.* Tioga Publishing Company: Palo Alto, CA.

Sleeman D. (1984) An attempt to understand students' understanding of basic algebra. *Cognitive Science*, Vol 8, Number 4, pp. 387-412.

Johnson W.L. and Soloway E. (1984) Intention-based diagnosis of programming errors. *Proceedings of AAAI*, pp. 162-168.

Stevens A., Collins A. and Goldin S. (1982) Misconceptions in students' understanding. In D. Sleeman and J. S. Brown (Eds.) *Intelligent Tutoring Systems.* Academic Press, London.

Trowbridge D. and Chiocariello A. (1985) Computer dialogs for developing concepts of physics. To appear in *Proceedings of the World Conference on Computers and Education*, Norfolk, Virginia.

VanLehn K. (1983) Human procedural skill acquisition: Theory, model and psychological validation. *Proceedings of the AAAI Conference.*

Wenger E. (1985) AI and the communication of knowledge: an overview of intelligent teaching systems. University of California, Irvine.

Winston P. (1980) Learning and reasoning by analogy. *CACM 23*(12), 689–703.

Winston P. (1982) Learning new principles from precedents and exercises. *Artificial Intelligence 19*, 321-350.

Winston P. (1983) Learning by augmenting rules and accumulating censors. *Machine Learning Workshop*, 2–11.