

searches. Wachsmuth postulates a *general* “middle-level” structure within which details can then be explored.

R.B.D.

## LETTER TO THE EDITOR OF JMB

In a line of thinking devoted to mathematical skills and skill automaticity, I found it helpful to think in terms of a metaphoric model that might be of interest to you and your readers. It views the human mind as a problem-solving system in the way artificial intelligence deals with it. Of course, this model is an oversimplification and cannot comprehend the many features of cognitive functioning in a problem solving situation. However, if only some features are captured by such a model, it might already prove a contribution toward a more profound understanding of the potentials involved in mathematics learning, and possibly provide a basis to express graphically some research hypotheses.

Let us think about an AI problem-solving system concerned with the solution of equations, for example, which is to solve a quadratic equation like  $x^2 - 6x + 5 = 0$ . Let us presuppose that the system is in a position to perform (“has automatized skills in”) arithmetical operations, term transformations (e.g., apply the binominal formula), equivalent transformations of an equation, etc. Let us further assume that the problem solving system “knows” when the application of such an operation is possible; that is, it is capable of recognizing that, say, a term is eligible for an application of the binomial formula. We shall have to be more concrete:

A problem-solving system consists, in principle, of three components (see, for example, the Handbook of Artificial Intelligence compiled at Stanford University):

- a database,
- a set of operators,
- a control strategy.

The database describes the task-domain. Initially, it holds the problem to be solved (here,  $x^2 - 6x + 5 = 0$ ), the goal (given by a description of the form of the solution), and factual knowledge relevant to the task (e.g., basic facts, associative law, the binomial formulas, etc).

The second component of a problem-solving system is a set of operators that are used to manipulate the database. In our case, it would contain operators for performing the basic computations, and term and equivalent transformations (e.g., apply a binomial formula, or, subtract the constant term at both sides). That is, the presupposed skills are part of the set of operators.

The third component of a problem-solving system is a control strategy for deciding what to do next—in particular, what operator to apply and where to apply it, in order to approach the goal. To this end, the control strategy can inspect both the database and the set of operators, and can access certain heuristics and tactics to find promising next operators to apply (we need not go into technical details here). An illustration of the structure of a problem-solving system is given in Figure 1.

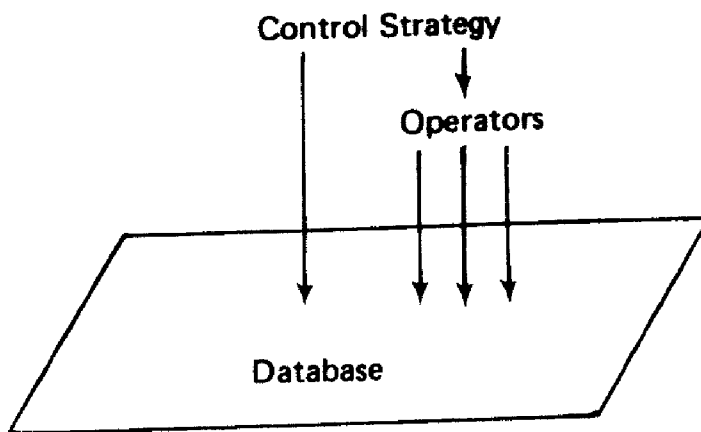


Figure 1.

In general, the object is to achieve the goal by applying an appropriate, finite sequence of operators to an initial task-domain situation. In our case this would mean to transform the initial equation,  $x^2 - 6x + 5 = 0$ , through a sequence of equivalent transformations until an expression is obtained which shows the solution for  $x$ ,  $x = 1 \vee x = 5$ . (We are not concerned here with the various AI techniques developed to *search* for such a "path".)

Suppose now that a problem-solving system *has* found an operator sequence appropriate for the solution of  $x^2 - 6x + 5 = 0$ , and suppose that the system is given another quadratic equation similar to the first one. Two things could happen.

Either, the control strategy could conduct another search in order to solve this equation, independent from the “past experience” with the first equation. Or, the control strategy could attempt to apply the same sequence of operators, *if* the successful sequence previously used has been recorded. (Certain AI systems are actually given this feature to add on operator sequences—“plans”—that have proven successful to their heuristics. This heuristic would then be chosen if a similarity between the two problems is “become aware of”, that is, identified in the task-analysis.)

If a certain operator sequence has proven to solve a certain *class* of problems, it can also be made a part of the set of operators in a problem-solving system (“learning an operator”; again, we skip technical details). For example, the sequence of operators achieving the solution of a quadratic equation by completing the square could become one “super-operator” (super-procedure). The applicability condition for this operator would be like “If the expression is of the form  $x^2 + c_1x + c_2 = 0$ , or  $x^2 + c_1x = c_3$ ”, and the action brought about by calling on this super-operator would consist of just the given sequence of operators. This is to say, the control strategy would now only be concerned with the recognition of whether an equation is of this particular form; once the according super-operator is brought to application, the whole sequence of its operators is applied to the task-domain situation, without further strategic control.

The analogy is now easily completed: The database in the problem-solving system could be thought of as the factual knowledge that is recollected from the memory of a person and put into her or his working memory, together with the problem to solve. (We do not worry about details here like that paper and pencil can be used to compensate for the fact that human working memory is limited in its temporal and spatial capabilities to retain information).

The set of operators are the rules, and skills, at one’s disposal. However, from the above it should be clear that only an *auto-*

*matized* skill can be viewed as one single operator which does not need strategic control *during* application.

And what is the control strategy in the problem-solving system, i.e. the "highest level" controlling and affecting both database and operators, can be viewed as the conscious, logical intellect. An illustration is given in Figure 2.

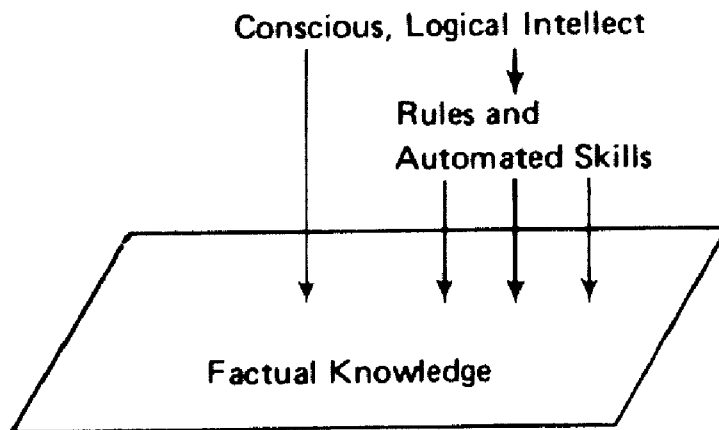


Figure 2.

Some implications of this model are near at hand now. For example, the difference between solving a quadratic equation meaningfully vs. performing the automatized algorithm is the following: The first case can be viewed as a problem-solving process, conducted under strategic control as an application of "more basic" skills and factual knowledge. The automatized algorithm, on the contrary, does not employ nor require such a reference to meaning; it is merely a routine procedure. In other words: In the first case the whole problem-solving system is occupied while in the latter the working memory is less occupied and the control strategy is free to do other possible things.

The transition of a meaningfully employed algorithm to an automatized skill of a person could be viewed as follows: An ordered sequence of operators which has proven successful is made to join the set of operators of a problem-solving system, as a super-operator. Rote-learning of a skill can also be viewed in terms of this model: It is subjoining ("programming") an operator into the human problem-solving system without having meaningfully coordinated the sequence of its sub-operators before.

It becomes apparent that access patterns and additional knowledge about when it is adequate to call for a particular procedure

have also to be learned, in order to achieve a reliable, alert interplay between control strategy and skill; especially when a skill is to be employed in a more complex situation. But it also becomes apparent that this is a business at a level different from learning a skill itself.

I have more in mind than is expressed in these brief notes, but I think it is worthwhile starting an overt discussion now.

Yours truly,

Ipke Wachsmuth  
Department of Mathematical Sciences  
Northern Illinois University  
DeKalb, Illinois 60115