Cognitive Theory to Guide Curriculum Design for Learning from Examples and by Doing

Xinming Zhu
Chinese Academy of Sciences, Beijing, P.R. China

Herbert A. Simon
Carnegie Mellon University, Pittsburgh, PA 15213

Dan Zhu
Iowa State University, Ames, IA

Yifei Lee
Chinese Academy of Sciences, Beijing, P.R. China

11 November 1998

1. Instruction

Designing curricula on the basis of fundamental cognitive theory is an aspiration of long standing. Within the past half century, we can call to mind Henry C. Morrison (1951), from the side of education, and Robert Gagne (1970), from the side of psychology, who approached the design task by analyzing the material to be learned into its unitary components. Similar analysis was undertaken by Skinner and his followers using the theory of operant conditioning.

In the past ten years, the development of computer technology and artificial intelligence has stimulated a new wave of activity in curriculum design for computer-aided instruction. Computer-based curricula take such diverse forms as intelligent tutoring systems (ITS) (Anderson, et. al., 1995; Nour & Younis, 1995), hypertext learning environments, including multimedia encyclopedias and textbooks (Raymond & Tompa, 1988; Jacobson et al. 1996), technology-supported learning communities (Warren 1997), and telematics for distance education (Oliver & Reeves, 1996).

In parallel fashion, modern cognitive psychology, especially within the information processing paradigm, has spurred research on curriculum design based on a variety of cognitive theories of learning (Rabinowitz, 1993; Elmore & Tennyson,
1993; Spada, 1993; Arruart, Fernandez & Greer, 1996). In particular, several research and demonstration projects have shown how students can acquire knowledge effectively from examples and by problem solving, using adaptive production systems as their model of knowledge and skill acquisition, and as the basis for designing the examples and problems used to implement the curriculum. Among early studies analyzing learning within the framework of adaptive production systems are Anzai & Simon (1979), Neves (1978), and Newell & Simon (1972). Two more recent examples, drawn from a much larger number, are Anderson et al. (1987a) and VanLehn (1987).

Some recent research on exploring human's learning and knowledge acquisition of mathematical or programming knowledge are also closely related (Klahr 1994, Siegler 1996, Koedinger et al. 1997).

In the same vein, Zhu and Simon (1987), with their associates in a Beijing public school, developed an entire three-year secondary school curriculum in algebra and geometry that enables students to learn from examples and "doing" (henceforth LFED), with a minimum of exposition and direct instruction. With this method students first acquire new productions by examining worked-out examples, then use them to solve new problems and receive feedback that produces further learning.¹

This curriculum is currently being followed by about 20,000 students in China. Instructional experiments have shown these curricula to be highly effective in comparison with traditional methods (Zhu et al., 1998, Zhu, Lee & Zhu, 1998). An experiment in teaching the principles of buoyancy in physics using LFED was equally effective (Zhu, Nan & Simon, 1994; Zhu, Lee, Simon & Zhu, 1996).

¹Various editions of these materials have been published in a number of volumes (in Chinese) as X. Zhu & H.A. Simon, Chuzhong shuxue shili yanbian jiaocai (Experimental materials for teaching by working examples in junior middle school.) Beijing, Chinese Academy of Sciences, Psychology Research Institute.
Various press articles reporting this study and results have appeared in the Chinese Educational Herald (the official Chinese Education Newspaper) and aroused tremendous interest. The instructors who got involved in different experiments also shared their confidence and experiences (A collection of these reports will be published in a book, I'll send these separately). In the meantime, exploratory experiments in teaching chemistry and other subjects are underway. The following table presents a comparison between the experimental groups and the control groups spreading over six middle high schools in six states (provinces) in the People's Republic of China. Results from these extensive experiments indicate that the scores of the students in this experimental class significantly surpassed those of their peers in the control group. (I'm not sure if we should include this much detail here, or shall we move this part to somewhere else?)
In this paper we describe how the theory of learning from examples can guide the design of curricula and discuss a number of principles of curriculum design derived from the theory. Perhaps we should call them "heuristics" rather than "principles," for they are intended as guidelines, and not as inexorable laws.

The principles themselves are mainly familiar, overlapping considerably with those that have been employed in other successful projects of this kind.\(^2\) It would be

\[^2\text{For example, five of the ten principles we discuss here are essentially the same as five of the eight discussed in Anderson, et al., (1995), and variants of the other three Anderson principles are included among our twelve sub-principles.}\]
worrisome if they were not, for that would imply that they were not very essential to
the effectiveness of the curriculum. Moreover, all the groups who work in this
domain draw on the same common body of theory and experiment.

Although the empirical evidence shows that well-designed curricula for
learning from examples are effective, other methods may be equally or more
effective under some conditions, and, given the difficulties of assessing educational
procedures, the efficacy of any method can, at best, be assigned only roughly to its
specific components. So this paper does not claim unique efficacy for a specific
method, but seeks to show how the theory of adaptive production systems provides
concrete practical advice for implementing this learning method effectively.

In instruction by LFED, a number of variant procedures have been proposed.
For example, although our approach and John Anderson's computer tutoring systems
(Anderson, et al., 1995) have common origins, we are led to somewhat different
prescriptions on various dimensions. For example, we place a greater emphasis on
learning the conditions of productions (cues) as the central learning objective, and
less emphasis on goal-driven action; and as a consequence, we encourage forward
search guided by relatively specific cues, as well as backward search guided by goals.
This emphasis reflects evidence that progress from novice to expert status is
associated with a gradual shift from working-backward methods to working-forward

We also do not, in general, introduce an initial stage of acquiring declarative
knowledge which must then be transformed into procedural knowledge, but design
the curriculum for direct acquisition of knowledge, even conceptual knowledge, in
the form of production rules. Some attention is given to helping students learn also
to express important knowledge declaratively, but we have not evaluated specifically
the usefulness of this for generalization and subsequent learning. The reasons for
these particular choices will appear as we proceed, but we should not like to claim that there is hard evidence for choosing among alternatives at this level of detail.

In the present paper, we will first introduce adaptive production systems and the methods of constructing and elaborating conditions for productions (CEC), and then discuss the principles of instructional design that follow from these methods and that have motivated our curriculum-building activities.

2. **Adaptive Production Systems and CEC**

Current cognitive theories postulate, with good empirical support, that the knowledge that enables skilled performance is stored largely as productions: if-then (or condition-action) statements consisting of a set of conditions followed by a set of actions, and usually designated C -> A. Whenever the conditions of a production are satisfied, the actions are evoked and carried out. A simple example of a production is: IF

1. The goal is to carry out arithmetic computations, and
2. There is a sequence consisting of a number followed by a plus sign
   followed by a number followed by an equals sign (e.g. 6 + 3 = );

THEN Write, to the right of the equals sign, the sum of the numbers that lie to the left of the equals sign.

A person who has this production stored in memory and is completing arithmetic equations (condition 1), upon reading or hearing such a sequence (condition 2), will add the two numbers (say, 6 and 3) and write the sum, 9. In this way, problem solving skills can be embodied in productions. In order to solve a problem, one must recognize the conditions that define the problem context and then execute the actions which are selected by these conditions. A production system is a set of productions of this kind, together with some rules for choosing which production to execute when more than one set of conditions is satisfied (conflict resolution rules).
Two central hypotheses provide the foundation for designing curricula based on the study of worked-out examples: (1) that human skills can be represented by productions, and (2) that these productions can be learned efficiently and with understanding by studying appropriate examples and/or by solving problems. A production system that can learn by modifying itself, altering its productions and adding additional ones to memory, is called an adaptive production system (APS).

2.1. The student described as an adaptive production system

The idea that a student can be described as an APS provided a new approach to the processes of human problem solving and learning, and to teaching problem-solving skills. Our task in this paper is to show how to identify the processes for learning to solve problems in a specific domain by specifying the production system that is to be learned. If we can specify such a system, then we can use it to design a series of problems and worked-out examples from which students can learn to solve problems in this domain.

An early application of this approach to school subjects was an APS using the LFED method, constructed by Neves (1978), which enabled a computer to learn how to solve linear equations in algebra. Shortly thereafter, Anzai and Simon (1979) used the Tower of Hanoi problem to explore further how APS's can, by solving problems, build productions that embody domain-specific knowledge. Zhu and Simon (1987, 1994) applied these ideas successfully to practical school instruction. Analyzing in detail students' processes of LFED in such fields as algebra, geometry and buoyancy they found that students not only learned to solve specific problems, but also acquired strategies and heuristic rules for transferring their skills to new problems.

Anderson and his colleagues (1983, 1985, 1987a, 1995), also taking production systems as models of students' skills, constructed CAI systems that were highly effective in teaching such subjects as geometry, algebra, and LISP programming. Basing their work on what they called Adaptive Control Theory (ACT), Anderson et al.
assumed that knowledge is first acquired declaratively, then converted into procedures by compilation. Thus, the students first learn verbal propositions, then transform them into skills in the form of productions — of perceiving cues and responding to them. According to this theory, a student would first learn the proposition:

"If the three sides of Triangle A are equal to the three sides of Triangle B, Triangles A and B are equal."

The student would then convert this to the procedure:

"IF sides a, b, c of Triangle A are equal to sides a', b', c' of Triangle B, respectively; THEN store assertion: 'Triangle A = Triangle B'"

Following Neves (1978), Anzai & Simon (1979) and Zhu & Simon (1987), we postulate that production rules can be acquired directly, without first learning their declarative equivalents. Our protocol analyses indicate that from the onset of learning, the processes students use for explaining examples and problem solutions by drawing upon previously-acquired knowledge are also used to acquire new domain-specific productions. In the succeeding problem solving, the students generalize the productions for broader application, and specialize them to handle special problem classes efficiently (Zhu, Nan & Simon, 1994; Zhu, Lee, Simon & Zhu, 1996).

Whether it is best to acquire declarative knowledge as an intermediate step before acquiring new productions or to acquire the productions directly is still an open research question. Our classroom experiments have shown that teaching the productions directly from examples is effective.

2.2 Extracting productions from examples

Solving linear equations in algebra illustrates learning a skill by extracting the requisite productions from examples. In this case, four steps are required (we assume that the student already is in the habit of simplifying algebraic expressions
when possible). The student has learned, just previously to encountering this example, that if the same quantity is added to or subtracted from an equation, or both sides are multiplied or divided by the same quantity (but not dividing by zero!), the solution of the equation will remain unchanged.

\[
\begin{align*}
(1) & \quad 17x + 14 = 9x + 30 \\
(2) & \quad 17x = 9x + 16 \\
(3) & \quad 8x = 16 \\
(4) & \quad x = 2
\end{align*}
\]

(Check: 17*2 + 14 = 9*2 + 30; 48 = 48)

The student already knows that the solution of an equation is of the form: \(x = N\), where \(N\) is some number that satisfies the original equation when substituted throughout for \(x\). Comparing line (1) with the solution, there is a 14 on the left side of (1) that is absent from the solution, and a 9x on the right side. Looking at (2) and comparing it with (1), the student sees that 14 has been subtracted from both sides of (1), removing one of the differences between (1) and the solution. The student (the APS) now constructs the production:

**IF** there is a \(N\) on the left-hand side of the equation,

**THEN** subtract \(N\) from both sides and simplify.

Now looking at (3) and comparing it with (2), it can be seen that 9x has been subtracted from both sides of (2), removing another difference. The student now constructs the production:

**IF** there is \(Nx\) on the right-hand side of the equation,

**THEN** subtract \(Nx\) from both sides and simplify.

Finally, looking at (4) and comparing it with (3), the students sees that each remaining term has been divided by 8, the coefficient of \(x\), removing the final difference, and providing the answer: \(x = 2\).

**IF** there is \(Nx\), with \(N \neq 1\), on the left-hand side of the equation,
THEN divide both sides by N.

A careful student will insert a fourth production of the form:

IF the expression is of the form, \( x=N \),

THEN replace \( x \) by \( N \) throughout equation (1),

and test for equality of the two sides.

Of course more than a single example is presented to the student to help infer the proper level of generality for the conditions of the rules and the actions to be taken. Then the student is asked to work some problems in which the solution steps are not given but must be supplied.

The basic process underlying the student's ability to discover and learn these new productions is *means-ends analysis* of the kind embodied in theories of human problem solving like the General Problem Solver (GPS) (Newell & Simon 1972) and Soar (Newell 1990). The final expression in the example, line (4) above, is the goal to be achieved; the first expression, (1), is the starting point. Each step reduces the distance between the initial expression and the goal expression, producing another expression on which the same process can be repeated until the goal expression is obtained.

Given the intermediate steps, the student doesn't have to discover the whole set of differences that have to be removed, but can deal with them one at a time by examining the changes from one expression to the next, and then retrieving from memory an operator that will reduce this kind of difference. This is the fundamental distinction between learning by doing and learning from examples. The presence of a sub-expression to be removed or altered becomes the condition side of the new production; the operator that effects the change becomes the action side; putting the condition side and the action side together creates the new production.

As the operator that is retrieved from memory, learned from previous instruction, is already generalized to any number (e.g., "Subtract N from both sides of