Techniques for Teaching Procedures

Brent G. Wilson
LEPS-GA 219
Northern Illinois University
Dekalb, IL 60115

Abstract. Procedures are among the most common kinds of content taught in many training programs. Complex procedures can be difficult to teach effectively. Several analytic methods for simplifying procedures for initial presentation are discussed. Most of the methods do not require analysis as thorough as an information processing task analysis, but they can still be useful to the practicing designer with limited time and resources.

Instructional designers and trainers face the problem of how best to teach different kinds of instructional content. In particular, procedures are commonly found in "how to" and technical courses. A *procedure* is a set of steps or performances to be carried out with the purpose of obtaining a predetermined outcome. Related terms include rule, algorithm, heuristic, and performance routine. Procedures may be tightly-defined or loosely-defined, simple or complex, certain or simply hopeful of obtaining the desired outcome.

Elaboration theory (Reigeluth & Rodgers, 1980; Reigeluth, Merrill, Wilson & Spiller, 1980) recommends teaching procedures through the use of an *epitome* which shows the key steps of a procedure, and allows the student to complete the simplest case in the first lesson of instruction. It is reasonable to suspect that since so many kinds of procedures exist, there may also exist a number of ways of formulating epitomes for those procedures. The purpose of this paper is to suggest several ways procedures can be introduced to students, depending on the kind of procedure and the purposes of instruction.

Path Analysis: An Obvious Parallel

P. F. Merrill (1976, 1978) and others (Scandura, 1973; Landa, 1976) have recommended that information processing principles be used in the analysis of instructional tasks. To do this, the task is analyzed to reveal the sequence of steps, including decisions and branches, carried out in its performance. A useful representation of this analysis is the flowchart, typically using boxes and arrows to designate different procedural steps and the order of their performance. An information processing (IP) task analysis represents an instructional task as a procedure.

According to P.F. Merrill (1978), a complete IP task analysis includes these general steps:

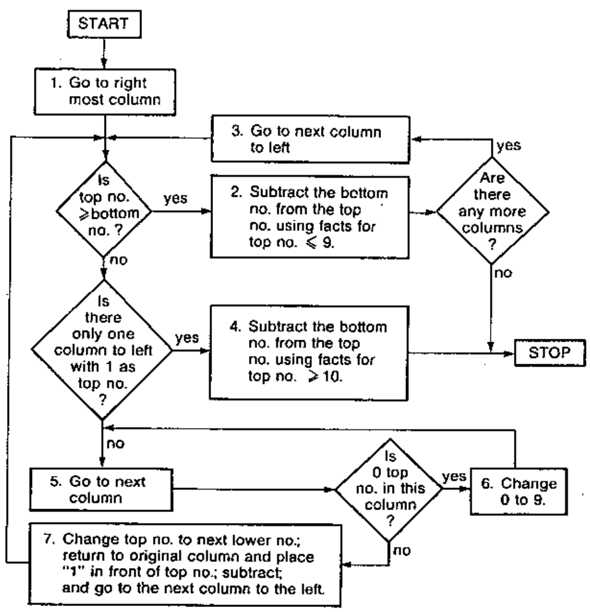
1. Identify the processing requirements of the task (the result is a precisely ordered series of operations and decisions, usually represented by a flowchart);
2. Identify the possible paths through the procedure; and
3. Order the paths according to difficulty (a path containing all operations of another path *and then some* can be said to be more difficult).

Figure 1 is an example of an IP task analysis. The flowchart representation is the product of the first step of analysis. The paths through the procedure (in this case a subtraction algorithm) show the different ways a student could perform the task. These paths are then ordered according to difficulty. Note that the number of operations contained in a path is *not* the criterion for difficulty; rather, difficulty is determined on the basis of one path's operations being *included* within another path's operations. It has been argued elsewhere (P. F. Merrill & Wilson, 1979) that IP task analysis identifies what elaboration theory intends by "epitome" for branching algorithms. Path analysis provides an efficient technique for carefully specifying what the core of the procedure is, that is, the simplest path through the algorithm. Following the path analysis from simplest path through more complex paths until the entire procedure is learned is consistent with the elaboration approach that starts with an epitome and gradually adds levels of elaboration (Reigeluth & Rodgers, 1980).

For more detailed exposition of the information processing approach to task analysis, see other sources (Reigeluth, 1983; Gagne, 1977). Path analysis is clearly an effective technique for sequencing procedures in instruction. It is important, however, to recognize the *limits* of path analysis as well as its strengths. Path analysis is only effective in prescribing instructional sequences when the procedure being used is (a) *well defined* (i.e., actually representing the processing requirements for performance) and (b) *branching* (i.e., containing decision points). When a procedure fails to meet either of these two conditions, path analysis techniques will be largely ineffective in determining how to teach the procedure. For example, some procedures are so grossly formulated that a single box of a flowchart may contain several sub-operations. Although path analysis may not apply, such "non-performable" procedures may still serve a useful instructional role. Take another case using a linear procedure, that is, a procedure without any decision points. Since only one path exists through the procedure, path analysis will be of no help in prescribing how to go about teaching the procedure. In the remaining portion of the paper, several techniques are presented for analyzing procedures.

1. Perform a path analysis. As described above, path analysis is assumed to be a first choice for designers and is performed whenever possible on a defined procedure.

2. Simplify the initial case. For branching algorithms such as that represented in Figure 1, path analysis can provide a partial ordering of the



Instances from different paths

PATH	INSTANCE
1→2	7 -3
1→4	13 -6
1→2→3	258 -13
1→2→3→4	153 -92
1→5→7→2	54 -27
1→5→7→4→6	1563 -875
1→5→7→2→3	268 -97
1→5→7→2→3→4	1663 -824
1→5→6→7→2	603 -578
1→5→6→7→2→3	4029 -3642
1→5→6→7→4	1300 -423
1→5→6→7→2→3→4	16059 -8797

Figure 1. Information processing analysis and illustrated paths (adapted from P.F. Merrill, 1978).

paths and suggest a possible epitome and sequencing of instruction. For algorithms or heuristics not so precisely formulated, it may be useful to simplify the procedure by choosing a simple first case. For example, the common variety of the "systems model" of instructional design usually contains no decision points but rather is intended as a general heuristic. A first attempt to actually use the model would be more successful if the development project were simple (e.g., a ten minute mini-lesson) than if the project were more complex (e.g., a six semester sequence of courses). The basic steps of instructional design could be carried out on a highly simplified task and thus provide students with an epitome for later elaborations.

Figure 2 illustrates a rather loosely constructed heuristic for literary criticism. There are several ways of simplifying this procedure for teaching. Reigeluth (1979) notes that one can simplify this procedure by using a short poem as the first case rather then a complex novel or play. The representation of the procedure remains unchanged, but the cases for which the procedure is used are kept simple for initial instruction.

3. Provide output for difficult steps. Another useful way of simplifying a complex procedure is to provide the output of steps that would otherwise have to be performed. This allows the student to perform critical parts of the procedure early on, as well as provide a meaningful context for the whole process. For example, statistics teachers often provide the values of sum of squares during tests, allowing the students to work with certain data as "given". This approach conceivably could be used before the student even knows how to calculate these values, if other parts of the procedure are considered more central or structurally important. The student can be taught these "subroutines" in subsequent instruction. This is similar to Gilbert's notion of "backward chaining" (Gilbert, 1962). In both cases, instruction allows the student to perform only part of the steps in a procedure, yet still benefit from seeing the logic of the entire procedure. In backward chaining, the student always completes the task; in our view, the student performs that part of the procedure that will highlight the logical structure of the procedure.

To take an example from office training, novices who first approach a word processing system will frequently be provided with a sample text file, thus avoiding the work required to enter, save, and load a body of text. The saving and loading of documents can be learned at a later time; with a text file in computer memory, students can master editing commands and functions.

There are four major stages in the multidimensional analysis and interpretation of creative literature:

1. Identify elements of dramatic framework—character, plot, setting, perspective, and language.
2. Combine the elements into composites appropriate for analysis of their literal meaning—(1) analysis of character, plot, and setting, (2) analysis of perspective, character, and plot, and (3) analysis of language.
3. Interpret the elements figuratively—symbolism through character, mood, and tone.
4. Make a judgement of work—personal relevance, universality.

Figure 2. Heuristic for analyzing literature (adapted from Reigeluth, 1979.)

4. Delete steps of the procedure. Another way to simplify the heuristic in Figure 2 would be to drop out or "prune" certain steps. Reigeluth (1979) suggests omitting consideration of "setting, perspective, and language" from the first stage of analysis ("identifying the elements of the dramatic framework") and only using "character and plot." This "pruning" technique can be carried on at all stages of this particular heuristic, resulting in a greatly simplified epitome that can be taught in a much shorter time. Of course, steps essential to the meaningfulness of the procedure cannot be deleted; the remaining procedure must still be sensible and suitable to the purposes of instruction. Whenever possible, a path analysis should be performed as an input to this strategy.

5. "Chunk" or summarize the steps. In some cases it may be profitable to teach a simplified heuristic as a conceptual overview rather than as a procedure to be performed by the student. As an introduction to a statistics course, for example, the teacher may want to show the general process of PROBLEM ANALYSIS→ DATA ANALYSIS→ CONCLUSIONS. Although this heuristic may not be operable in the first lesson, it may offer, nonetheless, a useful advance organizer for the course. This type of simplification is not a legitimate procedural epitome since it is not designed to be executed by the student.

The ways of simplifying procedures for instruction listed above should indicate the complexity of the problems of teaching procedures and the need for design flexibility in solving the problem. The challenge, of course, is deciding which technique to use and how to use it in a given situation. A heuristic to guide the designer's use of strategies is presented in Figure 3. This decision table lists conditions that may require different design strategies. The decision table takes into account procedures of varying complexity and varying purposes for presenting procedures. Path analysis is recommended when possible. Also note that "Use a simplified initial case" is recommended for all performable procedures, while "Delete steps for easier presentation" is recommended only for procedures too complex to present clearly in the first lesson.

6. Teach the principle(s) underlying the procedure. A final technique to be considered relates closely to M. D. Merrill's (1977) recommendation: "Identify the underlying principle" (p. 11). Elaboration theory suggests that the principle underlying a procedure should be used in deriving the epitome, but it does not specify how this is done. Bruner (1964) and Resnick (1976) have suggested that it may be important to teach the underlying principle *as well as* an efficient procedure, especially when the procedure is quite unrelated to the underlying principle (e.g., long division). Perhaps on occasion it would be helpful to the student if two separate procedures were taught, one that clearly illustrates the underlying principle, and another that offers an efficient performance routine. As a general rule, the greater the need for the student to be able to transfer the procedure to a variety of tasks and conditions, the greater the need to include some kind of "theory" in instruction. Where transfer is not an issue, there is less need to understand the underlying "whys" of a given procedure. Clearly the effort to make procedures meaningful is an area requiring further investigation.

Summary

A number of techniques for simplifying procedures for initial presentation have been described:

1. Path analysis. The different paths through a branching procedure are ordered according to difficulty. The simplest path through the procedure is taught first to the student, followed by increasingly difficult paths until the entire procedure is presented.

2. Simplify the initial case. Although the representation of the procedure and its steps may remain the same, a simple case may be used to demonstrate performance of the procedure in the first lesson.

3. Provide output for difficult steps. Rather than require that all steps of a procedure be performed, the teacher can provide the output of some steps for use in later operations.

4. Delete steps of the procedure. For some procedures, steps can be "pruned" while still preserving the meaningfulness of the procedure.

5. "Chunk" or summarize the steps. A complex procedure can be summarized and used as a synthesizer even in the first lesson. Presenting the procedure in this fashion, however, does not allow the student to perform the procedure immediately.

6. Teach the principle(s) underlying the procedure. This should be done as a supplement to teaching a specific performance routine. Understanding why a procedure works may improve the retention and transfer value of the procedure.

Most of the methods presented above are not nearly as clear-cut as path analysis. They have a practical value, however, to the designer whose resources may not always allow the expense of a thorough information processing analysis. It is helpful to know that a wide variety of techniques may be utilized in teaching procedures, all depending on the goals of instruction and the individual characteristics of the procedure and the students who must learn it.

Author Note. I wish to thank Harvey Black and Norm Higgins for helpful comments on earlier drafts of the manuscript, and to LuAnn Beasley for preparation of the manuscript.

SELECTING STRATEGIES FOR TEACHING PROCEDURES		
WHENEVER POSSIBLE:		1. Do a Data analysis.
IF, in the first lesson, the whole procedure:		THEN:
A. can be clearly presented and performed		2. use a simplified task as a first case (present whole procedure)
B. can be clearly presented but not performed		2. use a simplified task as a first case 3. provide output for difficult steps (present whole procedure)
	AND IF performance of the procedure is:	
C. cannot be clearly presented	intended	2. use a simplified task as a first case 3. provide output for difficult steps (if needed) 4. "prune" or delete steps for easier presentation
	not intended	5. "chunk" or summarize steps for easier presentation (do not require performance of procedure

Figure 3. Decision table to aid in selecting strategies for teaching procedures.

References

Bruner, J. S. (1964). Some theorems on instruction illustrated with reference to mathematics. In E. R. Hilgard (Ed.), *Theories of learning and instruction*. Chicago, IL: National Society for the Study of Education.

Gagne, R. M. (1977). Analysis of objectives. In L. J. Briggs (Ed.), *Instructional design*. Englewood Cliffs, NJ: Educational Technology Publications.

Gilbert, T. F. (1962). Mathetics: The technology of education. *Journal of Mathetics*, 7-73.

Landa, L. (1976). *Instructional regulation and control: Cybernetics, algorithmization, and heuristics in education*. Englewood Cliffs, NJ: Educational Technology Publications.

Merrill, M. D. (1977). Content analysis via concept elaboration theory. *Journal of Instructional Development, 1* (1), 10-13.

Merrill, P. F. (1976). Task analysis—An information processing approach. *NSPI Journal, 15* (2), 7-11.

Merrill, P. F. (1978). Hierarchical information processing task analysis: A comparison. *Journal of Instructional Development, 1* (2), 35-40.

Merrill, P. F., & Wilson, B. G. (1979, March). *Elaboration theory and information processing task analysis: A comparison.* Paper presented at the annual convention of the Association for Educational Communication and Technology. New Orleans, LA.

Reigeluth, C. M. (1979). In search of a better way to organize instruction: The elaboration theory. *Journal of Instructional Development, 2* (3), 8-15.

Reigeluth, C. M. (Ed.). (1983). *Instructional-design theories and models: An overview of their current status.* Hillsdale, NJ: Erlbaum.

Reigeluth, C. M., Merrill, M. D., Wilson, B. G., & Spiller, R. T. (1980). The elaboration theory of instruction: A model for sequencing and synthesizing instruction. *Instructional Science, 9,* 195-219.

Reigeluth, C. M., & Rodgers, C. A. (1980). The elaboration theory of instruction: Prescriptions for task analysis and design. *NSPI Journal, 19* (1), 16-26.

Resnick, L. B. (1976). Task analysis in instructional design: Some cases from mathematics. In D. Klahr (Ed.), *Cognition and instruction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Scandura, J. M. (1973). *Structural learning: Theory and research.* New York: Gordon and Breach.