The Practicality of Algorithms in Instructional Development

M. C. Mitchell, Jr.
Director of Instructional Development
Des Moines Area Community College
Ankeny, IA 50021

Abstract. Two of the most critical issues in instructional development (ID) are cost and time. Algorithmization promises to be a practical technology in optimizing instructional development, reducing costs, and saving development time. The notion of algorithms is briefly examined in this article. The differences between the pure, mathematical algorithms developed and applied in educational endeavors are discussed. Several algorithms employed in ID are spotlighted as examples of the varying roles they can take. The issues which set algorithms apart from other procedural forms are examined in light of available methodology for algorithms development. Some implications for developers of areas of application and research are noted. Finally, conclusions are drawn regarding the practicality of algorithms in instructional development.

More and more is being said and written about the use of algorithms in education. Landa's *Algorithmization in Learning and Instruction* (1974) served as a major stimulant to inquiry and discussion of the merits and purposes of algorithms. Merrill (Note 1), Gerlach and Brecke (Note 2), and Brecke, Gerlach, and Shipley (Note 3) have reviewed the literature on the educational applications of algorithms in the United States and Europe. By 1978, articles on the use of algorithms in education were appearing in the professional literature (Bung & Carrasco, 1977; Coscarelli, 1978; Gerlach & Hagin, 1977; Gerlach, Reiser, & Brecke, 1977; Kopstein, 1977).

Recently the use of algorithms as tools in instructional systems development (ISD) has begun to receive direct attention (Mager, 1973; Aagard & Braby, Note 4). However, those reports of attempts to algorithmize certain tasks of functions in instructional development (ID) have dealt only circumspectly with the practicality of algorithms in the applications. It is the purpose of this discussion to examine that practicality.

Algorithms Defined

Merrill (Note 1) reports numerous uses of algorithms in instructional development, particularly in the areas of task analysis and applying instructional strategies. However, although algorithms have a wide variety of uses and serve an important role in instructional systems development, they are not a panacea for all the problems encountered in the design of instruction (United States Army Adjutant General School, Note 5). According to Korfhage (1966), if a method is known for the solution, a problem need not be understood at all. However, in such a situation the method of solution must be specified exactly and in a language the user can interpret. Algorithms are exactly specified methods for problem solution which can permit educational practitioners to reliably perform certain ID tasks.

Lewis, Horabin, and Gane (Note 6) define an algorithm as an orderly sequence of instructions for solving a problem. Landa (1974) defines an algorithm as "a precise, generally comprehensible prescription for carrying out a defined sequence of elementary operations in order to solve any problem belonging to a certain class (page 11)." Gerlach, Reiser, and Brecke (1977) describe an algorithm as:

> . . . a procedure which possesses two attributes: generality and resultivity. If a procedure is applicable to a class of problems as opposed to a single problem, it is said to possess generality. If application of the procedure always leads to a correct result, the procedure is said to possess resultivity. (p. 14)

Knuth (1968) defines an algorithm in terms of five important features:

1. *Finiteness.* An algorithm must always terminate after a finite number of steps.
2. *Definiteness.* Each step of an algorithm must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case.
3. *Input.* An algorithm has zero or more inputs.
4. *Output.* An algorithm has one or more outputs, i.e., quantities which have a specified relation to the inputs.
5. *Effectiveness.* All the operations to be performed in the algorithm must be sufficiently basic that they can in principle be done exactly and in a finite length of time by a person using pencil and paper.

Markov (1962) lists the following three features as characteristic of algorithms:

1. *Definiteness.* The precision of the prescription, leaving no place to arbitrariness, and its universal comprehensibility.
2. *Generality.* The possibility of starting out with initial data, which may vary within given limits.
3. *Conclusiveness.* The orientation of the algorithm toward obtaining some desired result, which is indeed obtained in the end with proper initial data.

Korfhage (1966) presents another perspective on the definition of an algorithm. He generally defines a semi-algorithm as a procedure, used (rather) blindly to solve a problem, which succeeds after finitely many steps whenever the problem is solvable. He suggests that in addition an algorithm solves a related problem: Does the given problem have a solution?

Korfhage addresses some practical considerations for the use of algorithms. Although he writes of devices, meaning computer (electromagnetical) forms, one could substitute *instructional developer* for *device* without damaging the message. Making a substitution, the message is: The algorithm must be compatible with the [instructional developer]. That is, it must be state in a "language" which the [instructional developer] can comprehend, and must be a procedure which the [instructional developer] is capable of executing.

The preceding definitions clarified the theoretical construct of mathematical algorithm. However, Landa (1974) suggests that in education or the social sciences, the absoluteness of the mathematical algorithm is probably not attainable. He goes on to temper the rigid concept of an algorithm somewhat by defining what appears to be a more realistic concept—the *quasi-algorithm*.

Landa defines a quasi-algorithm as a prescription having a number of the essential traits of algorithms but lacking certain others, primarily univocality. Univocality exists when every user assigns the same label to an attribute or defines a concept the same way. Landa suggests that for further educational discussion the label algorithm, or algorithmic process, is acceptable if the term's limitations are understood—that univocality may not exist. Landa is not alone with his concept of a quasi-algorithm. Korfhage (1966) and Lewis, Horabin, and Gane (Note 6) refer to the concept as a *semi-algorithm*. Good (1973) and Merrill (Note 1) use the term *heuristic*. As a matter of fact, Good defines a heuristic as a procedure which may be compared with an algorithm yet may not always yield the desired output, thus defining the terms heuristic and quasi-algorithm (or algorithmic process) as synonymous. In an attempt to clarify the definitional issue, Brecke et al. (Note 3) proposed working definitions which distinguish quasi-algorithms from procedures and true algorithms.

While Landa described the differences between a mathematical algorithm and an algorithmic process in education at length, Good came right to the point and identified the important difference. An algorithmic process (heuristic), while possessing nearly all of the credentials of a mathematical algorithm, contains one significant flaw; that is the algorithmic process may *not* yield the desired result every time.

Why does the algorithmic process fail occasionally? Landa (1974) suggests two probable causes: the lack of univocality, and the difficulty of recognizing the attributes. While recognizing the second reason, he casts most of his support to the univocality issue, the attribute of the algorithmic process which others (Knuth, 1968; Markov, 1962) call definiteness or universal comprehensibility. Landa suggests that pedagogists (or educators) have some difficulty in defining the concepts they employ so that examples are always correctly classified.

> "An algorithmic process (heuristic), while possessing nearly all of the credentials of a mathematical algorithm, contains one significant flaw; that is, the algorithmic process may *not* yield the desired result every time."

As Landa (1974) pointed out, it is likely that many if not most algorithms or algorithmic processes applied in education or the social sciences would also be classified as quasi-algorithms. With that qualification in mind, then, what is the nature and future of algorithmic processes, or quasi-algorithms, as a practical technology in instructional design and development?

Examples and Current Applications

Use of algorithms in instructional design and development is expanding rapidly. Numerous examples of applications can be found in the literature. The Far West Laboratory for Educational Research and Development (Note 7) has devised an algorithm called ALERT (Alternatives for Learning through Educational Research and Technology) for selecting instructional materials to fill certain needs. Branson, Rayner, Cox, and Hannum (Note 8) developed an algorithm for reviewing and selecting existing instructional materials during instructional systems development. Merrill (Note 1) lists several examples of instructional design algorithms ranging from task analysis to prescribing instructional strategies.

Gropper (Note 9) employs an algorithmic approach to instructional systems development. Gropper's 11-volume ISD procedure appears to be the most complex and extensive algorithmic process in instructional development. Other examples of algorithms employed in instructional development are included in the U.S. Air Force's *Handbook for Designers of Instructional Systems* (Note 10), a 5-volume set, and Swezey and Pearlstein's *Developing Criterion-Referenced Tests* (Note 11). A handy algorithm for determining objective/test item match has been prepared by Mager (1973). Indeed, it is difficult to find recent complex ISD models that do not appear to employ an algorithm (quasi-algorithm) somewhere in their procedures.

Instructional management has provided fertile ground for the application of complex algorithms for resource management as well as record keeping and the management of instructional strategies. Two of the most impressive applications are the U.S. Air Force's Advanced Instructional System and the U.S. Navy's Systems Capabilities/Requirements and Resources Model (SCRR) (TAEG, Note 12). A portion of an SCRR algorithm is included here as Figure 1.

Hansen, Brown, Merrill, Tennyson, Thomas, and Kribs (Note 13) employed several algorithms in their adaptive instructional models (AIM). AIM was the theoretical framework for the U.S. Air Force's Advanced Instructional System (AIS). Twelve learning algorithms (instructional strategies) for 12 types of training objectives have been prepared by Aagard and Braby (Note 4). A portion of one such algorithm is illustrated in Figure 2.

A closer inspection of Aagard and Braby's learning algorithm for identifying symbols reveals these vital statistics:

- The algorithm is based on 11 guidelines derived from the literature on psychological learning principles.
- It contains 34 elements (attributes and operations).

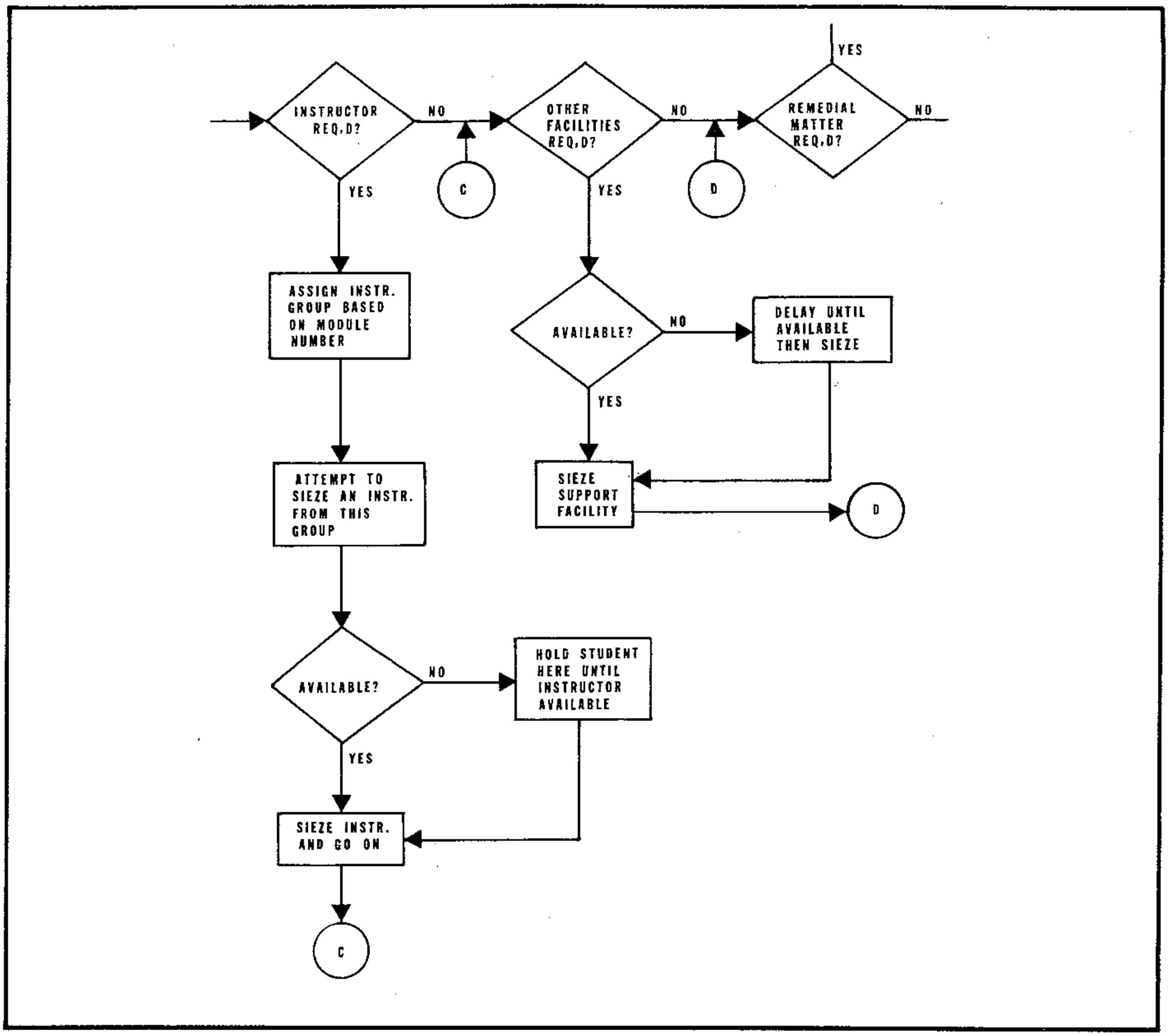

FIGURE 1. A portion of the Module Instruction and Execution Logic from the SCRR Model (TAEG, Note 12).

- Additional individualization of the instruction is provided by 11 loops.

It should be appreciated that the algorithm is complex; Figure 2 illustrates only a small part.

Earlier in this article it was pointed out that the algorithms applied in education were subject to occasional failure because pedagogists had difficulty in defining the concepts and principles they employ. The Aagard and Braby algorithm highlighted here (as well as the other 11 in the set) is based on such concepts and principles and therefore suspect.

In a test situation (Braby, Parrish, Guitard, & Aagard, Note 14), the algorithm was applied in the development of signal recognition training for the U.S. Navy. In this case it worked; validated instruction was developed. It is possible that the relative ease with which validation was achieved is attributable to the use of the algorithm, although there is currently no way to isolate that contribution. The validation issues will be discussed later.

Just how was the algorithm employed in the development? The learning algorithm for identifying symbols was employed as the instructional strategy. To use it the developers compiled a data base of symbols, referents, mnemonics, criterion test items, and suitable reinforcement statements. The algorithm specified the data base requirement and then, when the data base was in order, managed the instruction by controlling student movement throught the data base. The utility of the algorithm is that a proven strategy for instruction need not be "reinvented" for each development, and the algorithm defines the data base, thus clarifying the elements of the instruction which must be developed. The application of the algorithm during the development of instruction on identifying symbols reduces the time and effort involved in arriving at an instructional design. After all, the algorithm constitutes the design—and it is already developed.

Learning to identify symbols represents a very complex skill to be learned. This is the instructional arena in which the use of ID algorithms can make a significant contribution. Ob-

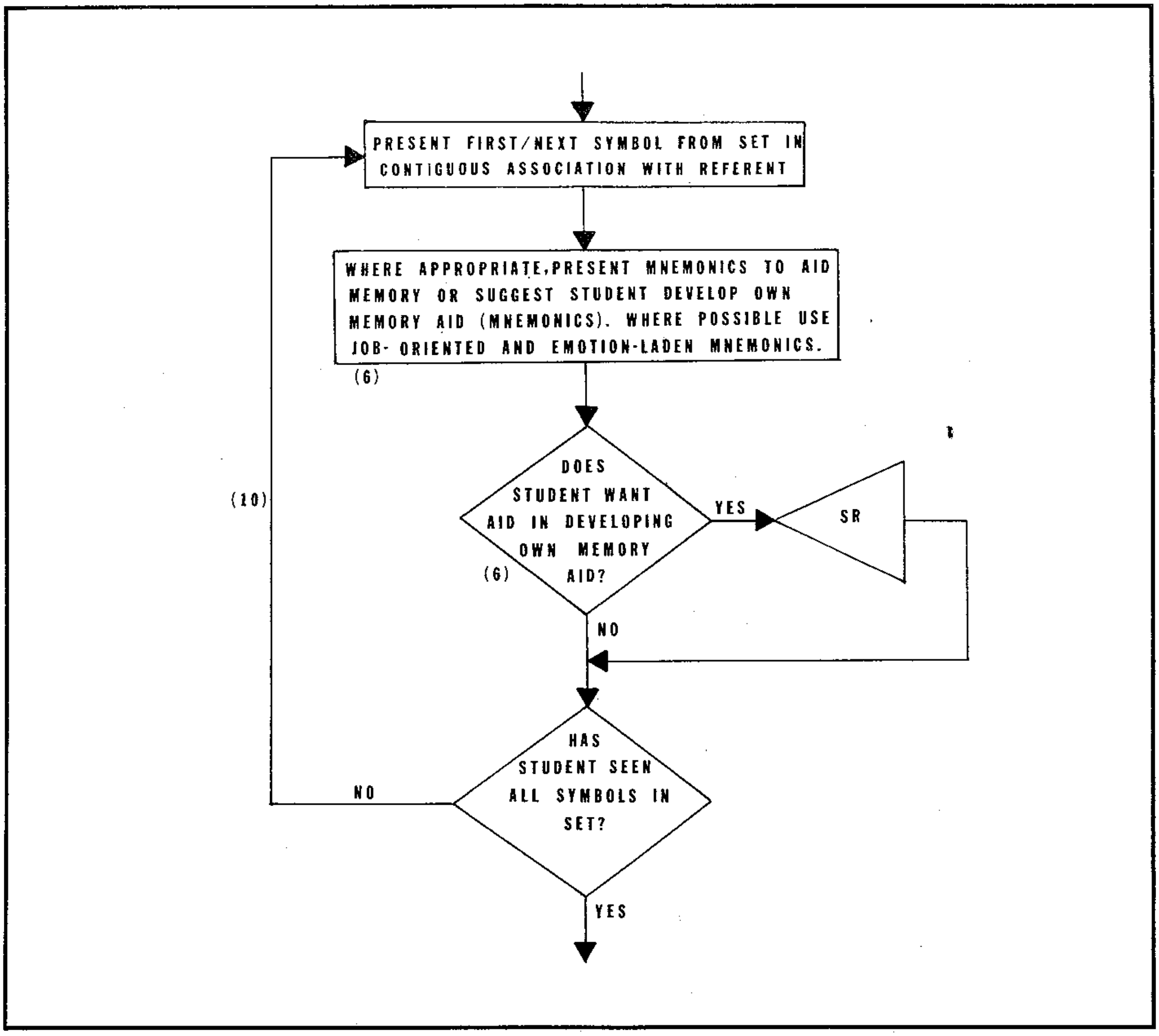

FIGURE 2. A portion of a learning algorithm for identifying symbols (Aagard & Braby, Note 4).

viously, once the algorithm is validated, it can be used over and over, requiring only the definition of the data base (examples, test items, rules, etc.) for the objective at issue. This is extremely important for a production-minded agency whose bottom line concern is the production of cost-effective instruction.

Issues of Development and Application

Although it appears that algorithms are being designed and applied in all phases of instructional design and development, few if any of those who have developed algorithms for instructional design and development have made any true effort to validate their contributions. As a matter of fact, from among all of the sources reviewed here, only one mention was made of validating instructional design algorithms—by Aagard & Braby (Note 4) regarding their 12 learning algorithms. They cautioned readers and users of their document that the 12 algorithms were *not* validated.

Several of the documents cited here for their use of algorithms were evaluated through formative trial (Branson et al., Note 8; USAF AFP 50-58, Note 10; Swezey & Pearlstein, Note 11). However, in each of these cases the evaluation involved the documents' ability to convey the processes and not the predictive validity of the algorithms themselves.

It can be concluded, then, that the practitioners of instructional design and development recognize the ability of algorithms, even quasi-algorithms, to produce fairly reliable results and to be both efficient and effective. Whether the apparent lack of concern about validation stems from a recognition of the difficulties surrounding validating algorithms or whether it stems from the belief of some practitioners that their algorithms are valid by definition, is not clear. What is clear is that either researchers in instructional design and development are not attempting to validate their algorithms, or they are not reporting their attempts.

Essentially, two types of algorithms are dealt with in instructional development. There are algorithms which operate as managers of instruction, such as instructional strategies and resources; the two algorithms in Figures 1 and 2 are of this type. There are also those which select or develop the instructional materials. The test item/objective match algorithm advocated by Mager (1973) is of this developmental type. The validity of the instructional algorithms can be verified directly through the performance of the learners. However, the validity of the developmental algorithms can only be inferred through the performance of the learners.

One of the most difficult issues in validating instructional development algorithms stems from the fact that the product of an algorithmic process (i.e., a test item) must be combined with other elements of the instructional system. Many developmental processes, algorithmic and nonalgorithmic, interact to develop the instructional system. This entire instructional system is then subject to validation.

Apart from the validation issue, the use of algorithms appears to be gaining popularity in instructional design and development as a practical technology. As attempts are made to translate the products of research in instructional design and development into field-useful concepts and rules, algorithms apparently provide a logical technology (Gerlach, Reiser, & Brecke, 1977).

The use of algorithms as the structure for many of the processes of instructional design and development may provide an excellent framework in which to carry out much needed developmental research. Designers are now beginning to react to the lack of firm research on the developmental aspects of systematic instructional design (Dick, 1977). For the most part, the various factors (or attributes) of systematic instructional design have not been systematically investigated. The use of an algorithmic structure in the design process not only permits such systematic investigation, it facilitates it. Merrill (Note 15) suggests that:

> . . . the effort to evaluate quasi-algorithms helps us to identify those steps which are ambiguous; which leads to the need to refine our concepts. It leads us to refine the definitions and identify a large enough set of examples and non-examples such that it is possible for others to classify examples and non-examples with greater precision.

Many instructional design principles and concepts, as well as other educational concepts, suffer from ambiguity which inhibits their effective use. Some of the principles heretofore granted unrestricted application in instructional design may have only limited usefulness. It is doubtful that the ambiguity can be entirely removed from some educational postulates such as the events of instruction (Gagné & Briggs, 1974). Such postulates form the cornerstone of modern instruction and as such may be the reason education is classified as an art rather than a science. Algorithmization of the designing of instruction can isolate these principles for systematic investigation and improvement.

The quasi-algorithm is certainly a viable alternative to the algorithm if two assumptions are recognized: that the general shortcoming of the quasi-algorithm (i.e., it does not always yield a correct result) is acknowledged; and that the application of the process can be validated so that it can be viewed as acceptably reliable. The first condition is simply a matter of acknowledgement. However, the second condition is not universally achievable.

> "Although it appears that algorithms are being designed and applied in all phases of instructional design and development, few if any of those who have developed algorithms. . .have made any true effort to validate their contributions."

Any further study of instructional development algorithms should include attention to the methodology employed in validating the application of the algorithms. The subjective elements should be reduced in favor of the objective elements. Horabin (Note 16) advocates an algorithmic process for validating algorithms. However, the process includes only the opinion of a panel of experts. Two major questions concerning the validation process should be asked: 1) How was the criterion correctness determined? and 2) Are the experts representative of the user population? If criterion correctness is based on opinion rather than definition, the best one could report would be that the application of the algorithm would have face validity. For instance, if the algorithm was intended for use in selecting instructional materials in a criterion-referenced instructional system, then to select materials simply on the opinion that they would be valid in the instructional system would set up a face validity situation. It would be much more meaningful if the application of an algorithm to some instructional development task had some predictive validity.

According to Nunnally (1967), "Predictive validity is at issue when the purpose is to use an instrument to estimate some important form of behavior, the latter being referred to as the *criterion*" (p. 76). With regard for the second concern stated above, then, if the subjects used in validating the algorithm are not representative of the user population, one could certainly question the algorithm's validity since it was not evaluated within its operational context. One could argue that training of users is really at issue here. However, we return to the criterion performance again: Was the criterion that a select expert could use the algorithm or that some other defined subject (or class of users) could use the algorithm? If one accepts Landa's (1974) condition of univocality, the training issue must precede the validation of the algorithm and only users can be employed in the validation study. There may be many acceptable procedures for establishing that the application of an instructional development algorithm is valid. Whatever the procedure employed, it should adhere to the principle that in order for the application of an algorithm to have predictive validity, the product of the algorithm must perform as intended.

Only when the methodological issues of validation have been resolved will instructional development algorithms be viewed by potential users with regard to their true usefulness. At present, users accept the algorithms they use because they *believe* they work better for them than any other technique they have reviewed. The decision to use any instructional development algorithm remains an intuitive rather than an objective one.

Implications for Developers

For those concerned with instructional development, the implications of algorithmization fall into essentially two categories: using algorithms and quasi-algorithms in instructional development and conducting research in instructional development with the aid of or in regard to algorithms.

Using algorithms and quasi-algorithms in instructional development allows the use of less highly skilled persons in ID. Algorithmization of certain tasks would permit many of the ID activities now performed by graduate level professionals to be performed by technicians with less sophisticated training. The realized personnel savings would certainly enhance the cost-benefit of the ID. In addition, algorithmization stadardizes both processes and the products of ID where standardization is desirable, and provides a way to insure high standards in developmental activities. Standardization is probably more important in large ID operations where technicians are being employed in certain tasks. When standardization is employed, day-to-day operational communications are simplified. As anyone who has worked in large-scale ISD knows, clear, simple communications are critical to efficient operation.

Research areas in the application of algorithms and quasi-algorithms include the general study of ISD and its functions, subfunctions, and elements. There appear to be four degrees of specificity in the systems approach to instructional development:

1. The ISD model or paradigm.
2. Procedures based on a chosen model.
3. Quasi-algorithms imbedded in, or instead of procedures.
4. Algorithms imbedded in, or instead of procedures.

Attempts at algorithmization in degrees 2 and 3 can assist in the improvement of instructional development by revealing issues of ambiguity and invalidity in currently accepted theory and dogma. Another area for research centers on development of algorithms for simulation, prediction, and management of ID resource allocation. This has particular import for large-scale ID operations and has already received some attention (O'Neal, Note 17). Finally, a fertile ground for researchers is the validity of Aagard and Braby's (Note 4) 12 learning algorithms.

Summary

Numerous attempts have been made to apply algorithmization to instructional systems development, thus saving time and reducing costs. However, the practitioner as well as the researcher should be cautioned against blind acceptance of published "algorithms," which are probably quasi-algorithms, meaning that they do not always yield the correct result. The issue of validation of instructional development algorithms is by no means resolved; however, if the ISD practitioner uses the algorithms with due caution distinct savings in time and expense may be realized. Additionally, the use of algorithms will provide an ideal framework for carrying out developmental research in instructional development.

> "Any further study. . .should include attention to the methodology employed in validating the application of the algorithms. The subjective elements should be reduced in favor of the objective elements."

Reference Notes

1. Merrill, P. F., *Algorithmic organization in teaching and learning: The literature and research in the USA.* Paper presented at the annual meeting of the American Educational Research Association, Chicago, April 1974.
2. Gerlach, V., & Brecke, F. *Algorithmic organization in teaching and learning: The literature and research in Europe.* Paper presented at the annual meeting of the American Educational Research Association, Chicago. April 1974.
3. Brecke, F. H., Gerlach, V. S., & Shipley, B.D., *Algorithms: A new tool for educational technology.* Paper presented at the annual meeting of the American Educational Research Association, Washington, D.C., March 1975.
4. Aagard, J. A., & Braby, R. *Learn-guidelines and algorithms for types of training objectives* (TAEG Report No. 23). Orlando, Fla.: U.S. Navy Training Analysis and Evaluation Group, March 1976.
5. U.S. Army Adjutant General School. *Systems engineering of training: Design and use of logic trees* (USAAGS Pamphlet 350-100-1), October 1969.
6. Lewis, B. N., Horabin, I. S., & Gane, C. P. *Flow charts, logical trees and algorithms for rules and regulations.* London: Her Majesty's Stationery Office, 1967.
7. Far West Laboratory for Educational Research and Development. *ALERT: Alternatives for learning through educational research and technology.* Berkeley, Calif.: Far West Laboratory for Educational Research and Development, 1970.
8. Branson, R. K., Rayner, G. T., Cox, J. L., & Hannum, W. H. *The instructional systems development model: Phase III–Develop.* Tallahassee, Fla.: The Center for Educational Technology, Florida State University, 1975.
9. Gropper, G. L. *A technology for developing instructional materials.* Pittsburgh, Pa.: American Institutes for Research, 1971.
10. U.S. Air Force. *Handbook for designers of instructional systems: Volumes I—V* (AFP 50-58). Washington, D.C.: Author, January 1974.
11. Swezey, R. W., & Pearlstein, R. B. *Developing criterion-referenced tests.* Reston, Va.: Applied Science Associates, 1974.
12. Training Analysis and Evaluation Group. *Design of training systems phase II report, volume III: Model program descriptions and operating procedures* (TAEG Report No. 12-2). Orlando, Fla.: Author, September 1974.

13. Hansen, D. N., Brown, B. R., Merrill, P. F., Tennyson, R., Thomas, D. B., & Kribs, H. D. *Technical report: The analysis and development of an adaptive instructional model(s) for individualized technical training*. Tallahassee, Fla.: The Center for Computer Assisted Instruction, Florida State University, 1972.
14. Braby, T. R., Parrish, W. F., Guitard, C. R., & Aagard, J. A. *Computer-aided authoring of programmed instruction for teaching symbol recognition* (TAEG Report No. 58). Orlando, Fla.: U.S. Navy Training Analysis and Evaluation Group, June 1978.
15. Merrill, P.F., Personal communication, January 30, 1978.
16. Horabin, I., *Algorithmic organization in teaching and learning: Practical application in education and training.* Paper presented at the annual meeting of the American Educational Research Association, Chicago, April 1974.
17. O'Neal, A. F. *Specification and development of computer aids to ISD.* Paper presented at the annual meeting of the American Educational Research Association, New York, 1977.

References

Bung, K., & Carrasco, M. J. S. Quasi-algorithms and the teaching of grammar. *Educational Technology,* 1977, *17*(10), 48–52.

Coscarelli, W. C. Algorithmization in instruction. *Educational Technology,* 1978, *18*(2), 18-21.

Dick, W. Formative evaluation. In L. J. Briggs (Ed.), *Instructional design: Principles and applications.* Englewood Cliffs, N.J.: Educational Technology Publications, 1977.

Gagné, R. M., & Briggs, L. J. *Principles of instructional design.* New York: Holt, Rinehart and Winston, 1974.

Gerlach, V. S., & Hagin. Algorithms for education and training. *Audiovisual Instruction*, 1977, 22(6), 46.

Gerlach, V.S., Reiser, R. A., & Brecke, F. H. Algorithms in education. *Educational Technology,* 1977, *17* (10), 14-18.

Good, C. V. *Dictionary of education.* New York: McGraw-Hill, 1973.

Knuth, E. E. *The art of computer programing, Vol. 1, Fundamental algorithms.* Reading, Mass.: Addison-Wesley, 1968.

Kopstein, F. F. Task specifications and diagnosis in educational technology. *Educational Technology,* 1977, *17* (10), 26–30.

Korfhage, R. R. *Logic and algorithms: With applications to the computer and information sciences.* New York: Wiley, 1966.

Landa, L. N., *Algorithmization in learning and instruction.* Englewood Cliffs, N.J.: Educational Technology Publications, 1974.

Mager, R. F. *Measuring instructional intent.* Belmont, Calif.: Fearon, 1973.

Markov, A. A. *Theory of algorithms.* Jerusalem: The Israel Program for Scientific Translations, 1962.

Nunnally, J. C. *Psychometric theory.* New York: McGraw-Hill, 1967.