

FOCUS ON EXCEPTIONAL CHILDREN

PROGRAM EVALUATION: NEW CONCEPTS, NEW METHODS

Gary D. Borich¹

PROGRAM EVALUATION

The failure of many efforts to evaluate educational programs stems from a lack of understanding of the programs themselves. An *educational program* is a set of hierarchically arranged instructional experiences that interrelate to generate several well-defined terminal outcomes. The purpose of *program evaluation* is to revise, delete, modify, add to, or confirm the efficacy of these experiences.

The key to understanding how and why a program brings about the outcomes it does lies in that program's hierarchical structure, or the way in which its components build upon one another to achieve outcomes greater than those that can be expected from any single part. It is the program evaluator's understanding (or misunderstanding) of this systematic interrelationship of components that often determines the utility and relevance of the evaluation to program developers and implementers. When evaluators fail to base their evaluation designs on a thorough understanding of program purpose and organization, their results and conclusions seldom address the needs which prompted the evaluation. Since their results and conclusions fail to integrate existing conceptions of the program, they cannot provide direction for program revision or modification.

Need and Purpose of Decomposition

A "components" view of an educational program assumes that behavior is generated or changed by specific, discrete instructional activities, and that the interrelationships among these activities build to more general behaviors at program completion. In any large-scale program that encompasses an almost endless array of instructional experiences, some activities can be expected to benefit program participants, some to hinder program participants, and still others to have no measurable effect upon them. The purpose of program evaluation is to assess the instructional activities that comprise the global program in a manner that makes possible the rendering of a judgment as to whether these activities should be revised, deleted, modified, unchanged, or supplemented with additional instructional components.

The role of program evaluation, then, is not only to decompose the program and,

1. Dr. Borich is Director of the Instructional Systems Laboratory and Associate Professor, College of Education, The University of Texas at Austin. Portions of this work were supported by the National Institute of Education, NIE-C-740088.

hence, understand the nature of its parts, but also to collect evaluative data from which to judge the adequacy of each component. Should a program component fail to engender the intended outcome or to relate with other program components to produce more comprehensive program outcomes, the effectiveness of that program component can be questioned.

Generally, program evaluation has placed little emphasis on the hierarchical nature of the program and, thus, has often failed to address the program's generic purpose—to gradually build more complex outcomes through hierarchically arranged instructional experiences.

Underlying the concepts presented in this article is the belief that program evaluation cannot be divorced from program definition, that evaluation functions not only after but also *during* program development, and that the evaluator cannot judge a program's parts without considering the composition of the whole. In the concepts that follow, the evaluator will be seen not only as an analyzer of data and reporter of program effects, but also as a logician and systems analyst. This perspective differs from typical notions of the role and function of the evaluator. While traditional representations of the evaluator are not entirely invalid, they often portray the relationship between the evaluator and program planners, designers, and developers as limited and distant. Such a relationship allows the evaluator minimal exposure to the program in its earliest stages and affords him little opportunity to assist

planners, designers, and developers in fostering a common conceptualization of program components and their interrelations.

The traditional view of program evaluation is best illustrated by arranging the role functions of those involved in program planning, development, and evaluation on a single continuum, as shown in the top half of Figure 1. Traditionally, *program planning* has included two roles: the planner and the designer. *Program development* has involved the roles of developer and formative evaluator. And *program evaluation* has included both the formative and summative evaluator, the latter being primarily responsible for comparing the program with a control or alternative program.

Unfortunately, such a continuum of activities and role functions separates planning and development activities from what are seen as legitimate evaluation activities. This distinction between developer and formative evaluator has prompted many to view the two as opposing forces, some arguing that the formative evaluator must guard against the influence of the developer who is likely to be favorably biased toward the program, and others arguing that the developer and formative evaluator must work in close relationship in order to achieve the best mix of evaluation and development.

In practice, this conception of role functions often encourages the emergence of formal boundaries between program planning, program development, and program evaluation. Where one activity ends, the next begins with a different set of tools and techniques. Thus, it is not uncommon to find planners, designers, developers, and evaluators each beginning their work with a different "picture" of what the program is supposed to accomplish. While the professional boundaries generated by these roles may be inevitable with a compartmentalized view of planning, development, and evaluation, we need not persist in maintaining this conceptualization with the emergence of a methodology that allows us to link these role functions. While some insights have been gained into the evaluative process by these highly specialized roles, any further division of the evaluator from the developer and planner may not be in the best interest of program planning, development, or evaluation.

The bottom portion of Figure 1 presents a second continuum, on which the evaluator is shown contributing to initial planning and development. Here the evaluator, rather than entering the scenario late in the development process, plays an integral role in program planning and development *alongside* planners and developers. What are the evaluator's functions in this new role?

The revised role of the evaluator as depicted in the

FOCUS ON EXCEPTIONAL CHILDREN is published monthly except June, July, and August as a service to those concerned with mentally retarded and emotionally disturbed children. This journal is abstracted and indexed in *Exceptional Child Education Abstracts*, and is also available in microfilm from Xerox University Microfilms, Ann Arbor, Michigan. Subscription rates, \$10.00 per year. Copyright 1977, Love Publishing Company. All rights reserved. Reproduction in whole or part without written permission is prohibited. Printed in the United States of America. Second class postage is paid at Denver, Colorado.

Executive and Editorial Office
6635 East Villanova Place
Denver, Colorado 80222
Telephone (303) 757-2579

EDITORIAL BOARD

Edward L. Meyen
University of Kansas

Glenn A. Vergason
Georgia State University

Richard J. Whelan
University of Kansas Medical Center

Carolyn Acheson
Senior Editor

Stanley F. Love
Managing Editor and Publisher

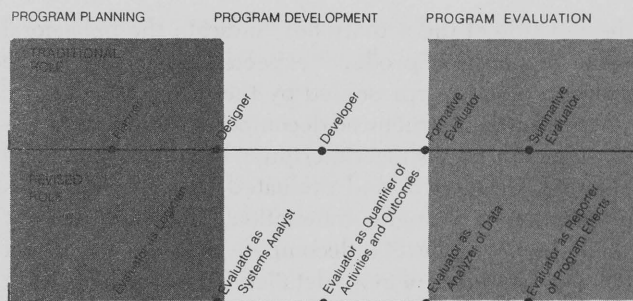


FIGURE 1

Some Old and New Conceptions of the Evaluator

figure demands a technique that can be applied throughout the planning, design, development, and evaluation process to define and describe the program, to clarify its purposes and intents, and to foster a common conceptualization of it among project personnel. Ideally, such a technique would provide a basic language to use in articulating the program during all stages of planning, design, development, and evaluation. It also would allow the evaluator to serve as *logician* and *systems analyst* in order to clarify and focus the work of the planner and designer, as *quantifier of program activities and outcomes* to provide data for analysis, as *analyzer of data* to determine program effects, and as *reporter of program effects* to communicate results and conclusions.

Traditionally, program development and evaluation have been viewed as two distinct roles or functions, related in sequence but not substance. Formal training in instructional design and development rarely includes evaluation concepts and vice versa. While the notion of formative evaluation may link development and evaluation in theory, it has in practice failed to achieve significant symbiosis between these two activities. Given the nature of program development, it seems strange that it has taken evaluators so long to pose a closer relationship among evaluation, development, and planning under the aegis of formative evaluation. It is equally strange that in the 10-year period since the concept of formative evaluation has been articulated, the concept has produced so few tools by which the evaluator can relate to, understand, and communicate the nature of a developing program.

THE NATURE OF DECOMPOSITION

Structured hierarchical decomposition is a technique

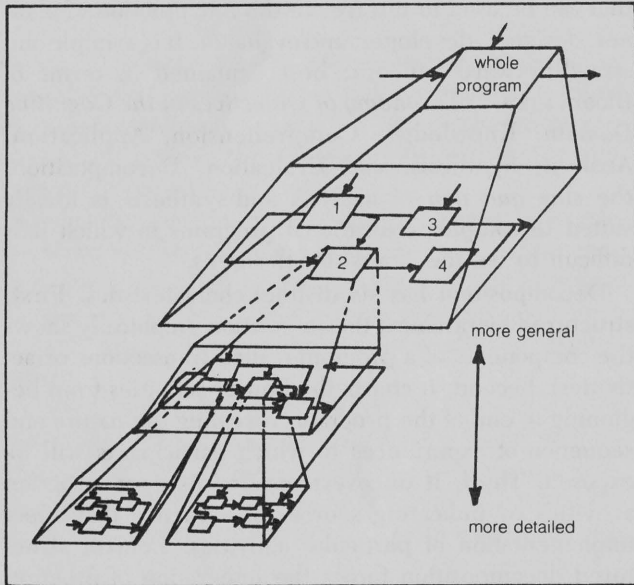
that can be used to interrelate the role function of planner, designer, developer, and evaluator. It is a simple and straightforward concept, best explained in terms of Bloom's (1972) *Taxonomy of Objectives in the Cognitive Domain*: Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation. Decomposition, the *sine qua non* of analysis and synthesis, is ideally suited to complex educational programs in which it is difficult to see the "forest for the trees."

Decomposition has six distinct characteristics. First, structured hierarchical decomposition graphically shows the components of a program (called transactions or activities). Second, it charts the flow of activities from beginning to end of the program, revealing the nature and sequence of experiences to which participants will be exposed. Third, it uncovers constraints upon program activities by indicating sources of influence that affect implementation of particular activities. Fourth, structured decomposition forces the integration of program parts by simultaneously detailing both the activities to be provided and the behavioral outcomes to be expected. Fifth, it fosters a common conception of the program by providing planners, designers, and evaluators the opportunity to work in team-like fashion on the decomposition and modeling task. And, sixth, structured decomposition builds for planners, designers, and evaluators a working vocabulary with which to describe key concepts in concise semantic and graphic terms for use across the planning, design, development, and evaluation phases of the program.

Structured decomposition achieves the above objectives by depicting the program hierarchically, in top-down fashion. Program detail is introduced gradually so that substantive detail is integrated into the whole without obscuring the overall intent or "big picture."

Figure 2 presents the decomposition process by showing program activities as boxes and the outcomes or "data" expected from these activities as lines connecting the boxes. When a program activity is decomposed into subactivities, interfaces among subactivities are shown as arrows. The title of each subactivity along with its interface arrows circumscribes a context in which program planners, designers, developers, and evaluators can work in detailing the precise nature of that subactivity.

A typical procedure is to evaluate a program by focusing on its sequence, beginning with activities on day 1 and following through to day *n*. This practice can unnecessarily confine the evaluator's understanding of the program to lateral flow. Hierarchically organized transactions and results can evade the myopic view of the development



Each module in a model is shown in precise relationship to other modules by means of interconnecting arrows. When a module is decomposed into submodules, all interfaces between the submodules are shown as arrows. The title of each submodule plus its interfaces define a well-constrained context for the detailing of that submodule.

FIGURE 2
Decomposition Process

team, no matter how thorough their efforts to uncover all program activities and planned outcomes. In such a case, the problem is often not with the specialists who are attempting to define and describe the program but with the one-dimensional model they are using to evaluate it. Structured decomposition keeps the hierarchical intent of the program in full view while gradually introducing substantive detail, using input, output, and control arrows to relate activities at a given level to those at any other level.

Program transactions are brought to life via inputs, controls, and outputs which lead to or emanate from each activity box. Inputs, always positioned on the left side of an activity box, represent raw data (e.g., participants, materials, processes) which stimulate the transaction and are eventually converted to output, or "changed" participants, materials, or processes. Control data, always indicated by arrows at the top of the activity box, indicate how the input may be constrained (e.g., by \$), regulated (e.g., by policies), or modified (e.g., by knowledge of the quality of output) to produce the output. And, output data, shown by arrows emanating from

the left side of the activity box, indicate the behavioral effect or finished "product" expected as a result of the program activity represented by the box.

Application of structured decomposition starts with the most general or abstract description of the program to be planned, developed, and evaluated. If we confine this description to a single "transaction," represented by a single box, we can then decompose or break down that box into a number of more detailed boxes, each of which symbolizes successively more detailed program activities. Each of these more detailed boxes can be further decomposed to amplify information contained in the parent boxes. This top-down approach thus avoids the complication of considering too many details too soon by introducing substantive detail gradually, and in meaningful steps, to form an overall picture of the interrelationships among program transactions.

Today, most organizations and agencies communicate planning and design decisions to program developers with a program proposal which describes the planned activities and their intended effects. The proposal is often the most formal expression of program intent. Not coincidentally, it is usually the only document available to aid program evaluators in selecting and guiding their own activities. Because the proposal must often respond to political as well as substantive considerations of the funding agency, it frequently provides only a broad, global description of program components, unified by loose scaffolding upon which global program objectives must often be supported. Thus, the proposal rarely serves program developers and evaluators as a definitive guide to intended transactions and expected outcomes. Hence, not only are evaluators uninvolved in the development process, but developers themselves are often unsure of program intents since planning and design decisions are poorly communicated from program designer to program developer. Clearly, there is a need for a systematic methodology by which to transmit planning and design decisions to developers and to systematically define, focus, and refine program transactions and intended outcomes *prior to* formative evaluation. It seems only natural that the evaluator should assist in the early articulation of program concepts by serving as logician and systems analyst during the planning and design phases. A systematic methodology for accomplishing this purpose is structured hierarchical decomposition.

STRUCTURED DECOMPOSITION DEFINED

The idea that the human mind can understand any

amount of complexity, as long as it is presented in small, accessible chunks that are linked together to make the whole, is the basic assumption of structured decomposition. For the past several years computer software development specialists have been developing, applying, and improving general but practical approaches to handling complex system problems. The approach taken in this article borrows heavily from the work of Douglas T. Ross (1977), which has become known as the Structured Analysis and Design Technique (SADT)*, one of a family of structured decomposition techniques. The basic ideas of these computer software specialists, however, are applicable to any field in which there is a need to effectively communicate the interrelationships among activities and outcomes occurring in complex systems or programs.

In the area of software computer technology, the application of decomposition methodology to real-life environments has significantly increased the productivity and effectiveness of teams of specialists involved in a development project (Ross & Schoman, 1977). The structured decomposition approach provides methods for thinking in an organized way about large and complex programs, for working as a team with effective division and coordination of effort and roles, and for communicating planning, development, and evaluation decisions in clear and precise notation.

The following fundamental assumptions underlie the application of structured decomposition to program evaluation:

1. Programs are best studied by building a model which expresses an in-depth understanding of the program, sufficiently precise to serve as the basis for program development.
2. Analysis of any program should be top-down, modular, hierarchic, and structured.
3. Program activities should be represented by a diagram which shows components, their interfaces, and their place in the hierarchic structure.
4. The model-building technique must represent behaviors to be produced, transactions to be provided, and relationships among these behaviors and transactions.
5. All planning, design, development, and evaluation decisions must be in writing and available for open review by all team specialists.

Structured decomposition uses a model to define the program. As indicated, this modeling process may be

applied to a variety of programs, whether or not they are highly structured.

Structured decomposition systematically breaks down a complex program into its parts, henceforth called instructional transactions. Structured decomposition starts with a general or abstract description of the program, which serves as a working model from which successively more detailed portions of the program are conceived. Graphically, this process involves division of a cell representing the overall program into a number of more detailed cells, each symbolizing a major transaction within the parent cell. The extent of analysis within any step of structured decomposition is limited to a small number of transactions, each of which is further broken down in succeeding steps of the process. This approach ensures uniform, systematic exposition of successive levels of detail.

Because the complex interrelationships among program activities do not lend themselves to clear and concise expression in prose, structured decomposition utilizes a graphic language designed to expose detail gradually in a controlled manner, to encourage conciseness and precision, to focus attention on module interfaces, and to provide an analysis and design vocabulary for use by program planners, developers, and evaluators.

In summary, structured decomposition is a methodology which can be used by planners, developers, and evaluators for:

- thinking in a structured way about large and complex programs;
- communicating planning and design concepts to developers and evaluators in clear and precise notation;
- insuring the accuracy, completeness, and quality of an evolving program description with procedures for review and approval;
- documenting program evolution, planning and design history, and related decisions;
- working as a team with effective division and coordination of effort;
- managing and guiding the development of a project; and
- providing strategic concepts for assessing the results of the planning, designing, and development process.

From an adequately constructed decomposition of the program, it should be apparent why the program was created and the technical, operational, and economic considerations that provide criteria for the various component parts of the program; what the program is to be in

terms of its specific components and activities; and how the planned program is to be constructed and implemented.

BEGINNING THE EVALUATIVE PROCESS: USING THE DECOMPOSITION

From an examination of the decomposition of program activities and outcomes, evaluators often can suggest program modifications on logical grounds. Planners, designers, developers, and evaluators all use the decomposition model to interpret the program's meaning and to bring all parties who have a stake in the program into agreement about its intents and purposes. The heart of the decomposition is program transactions. The evaluator, in particular, uses decomposition to identify incongruencies between transactions and outcomes. Many times these incongruencies, unnoticed with lateral flow decomposition techniques such as PERT diagrams, appear so obvious with hierarchical decomposition that program development is halted until logical relationships between transactions and outcomes can be achieved either by redefining behavioral expectations or revising the nature of program transactions. In addition, program evaluation efforts may be temporarily shifted to redefinition of ambiguous parts of the program and construction of better, more effective transactions and more realistic outcomes.

Several concepts can help the evaluator affirm the logic of the relationships mapped by hierarchical decomposition. These concepts serve as primary links in the process of uncovering mismatches between program intents and program transactions and between program transactions and expected outcomes. To affirm that logical relationships among these are in evidence from the decomposition diagrams, the evaluator identifies the level of inference of each program transaction and classifies each output as a terminal or enabling behavior. Here is how the process works.

Transactions that are directly related to the behavioral outcomes expected of participants at program completion (let's call them terminal outcomes) are considered low-inference: We can infer that completion of the transaction will improve performance on the terminal outcome. The transaction may even require the participant to perform a portion of the behaviors that are expected at program completion. In such a case we say the fidelity of the transaction is high. The judgment that successful completion of the transaction by program participants will lead to improvement in their terminal behavior is small (low

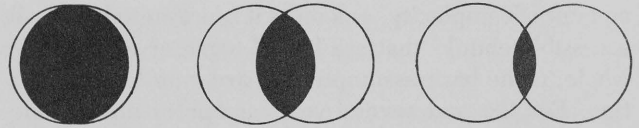


FIGURE 3

Degrees of Fidelity between Instructional Transaction and Terminal Outcome

inference).

Figure 3 illustrates three conditions of fidelity between program transaction and terminal outcome. In the first instance, the overlap between the behavior produced by the transaction and the type of performance expected at program completion is almost complete. Here, fidelity is high: One need make only a small inference that if the transaction is successfully completed, terminal performance will improve. In the second instance, some fidelity is apparent, but the overlap is not nearly as great as in the first case. This transaction would be called medium-inference. In the third example, only a small portion of the behavior expected as a result of the transaction matches that which is expected at program completion and, thus, the transaction is one of high-inference and accordingly has low fidelity with the terminal behavior.

Not surprisingly, low-inference transactions are linked to terminal outcomes by relatively few enabling behaviors. Their relationship to the behaviors expected of participants at program completion is direct and uncluttered by many mediating processes.

On the other side of the coin are high-inference transactions which, due to their low fidelity to terminal behaviors, must be connected to the latter via many mediating processes and clustered with other transactions before their impact on the terminal performance of program participants can be measured. This relationship is illustrated in Figure 4.

Note that at some point along the curve in Figure 4, the curve flattens and the number of mediating processes required to link the transaction to the terminal behavior may exceed available resources. Transactions at higher levels of inference may not be cost-effective. Both low- and high-inference transactions are important ingredients in program composition, and developers must not favor one over the other. Typically, high-inference transactions comprise orienting or introductory activities, such as reading a chapter in a book, listening to the teacher lecture, playing a recording, etc. Each by itself is likely to improve a terminal outcome (e.g., reading at grade level) only slightly, if at all. While the fidelity of these transactions to terminal program outcomes may be low, they

nevertheless may be prerequisite to a long sequence of transactions and enabling behaviors which together comprise a significant and necessary portion of the program. Low-inference transactions, on the other hand, are directly related to terminal outcomes and may actually require program participants to perform all or a significant portion of the behaviors expected at program completion.

Figure 5 (pp. 10-11) depicts the structured hierarchical decomposition of a program designed to prepare regular preservice teachers for mainstreaming. Try to identify its low-inference and high-inference transactions by noting the relative proximity of the instructional activities shown in the boxes to the activities the trainees are expected to perform at program completion. Remember higher-inference transactions commonly comprise more global, orienting or introductory experiences that are prerequisite to lower-inference transactions which approximate the real-life tasks for which training is being provided.

The concepts of high- and low-inference help the evaluator to determine gaps or mismatches between program intents and program transactions and between program transactions and program outcomes. Often program planners and developers make what evaluators call "inferential leaps" by espousing certain objectives for a program, but failing to provide the resources or specifications by which to incorporate the required transactions into the program at the appropriate level of inference. Or, transactions can be mismatched to outcomes in a similar manner.

For example, (1) high-inference transactions are sometimes expected to produce behaviors which approximate

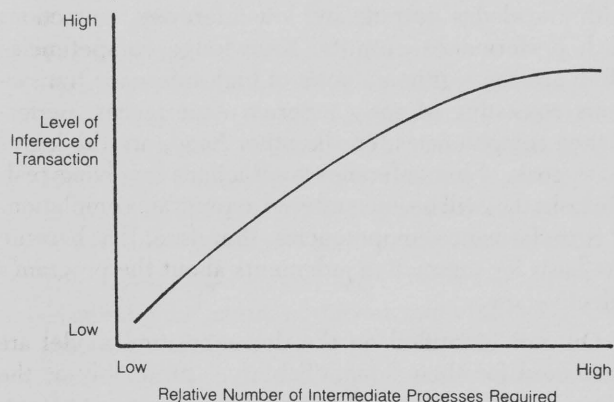


FIGURE 4

Generalized Curve: Relative Number of Intermediate Processes required for Transactions at Different Levels of Inferences

terminal outcomes, or (2) low-inference transactions are sometimes expected to produce enabling behaviors which may have little or no relation to terminal outcomes. Just the opposite should be noted on the decomposition model. In the second type of mismatch noted above, the cost-effectiveness of the match-up might be questioned. Low-inference transactions usually are costly to develop because their purpose is to produce *performance* as well as to instill *knowledge*; thus, the desired enabling behavior could be attained in a more efficient and less expensive manner. Such mismatches are exposed by decomposition diagrams since transactions (shown as boxes) and expected behaviors (shown as output arrows) are contiguous.

Table 1 provides readers with an opportunity to practice matching transactions and outcomes. The answers to this exercise can be found at the end of this article.

TABLE 1
Matching Transactions to Outcomes

Choose the most appropriate outcome on the right for each transaction on the left.

TRANSACTION	OUTCOME
— reading about a concept	1. evaluation; i.e., decision-making, appropriately judging or selecting the concept in an ongoing setting
— writing or completing exercises about the concept	2. application; i.e., using the concept in a situation different from the one in which it was learned
— practicing the concept in a simulated environment	3. comprehension; i.e., translating the concept into different terms, summarizing it, organizing it differently
— using the concept in a real-life (performance) setting	4. knowledge; i.e., recognition and recall of facts, defining terms, recalling names

BEHAVIORS, VARIABLES AND COMPETENCIES

Many of the ambiguous findings produced by evaluation studies can be traced to poorly defined outputs. Outputs can be expressed not only as enabling the terminal behaviors but also as behaviors, variables, and competencies. It is important for evaluators as well as designers and developers to note the distinction among these concepts.

The term *behavior* involves the most general level of description. The meaning of a behavior often is conveyed by relating it to other constructs with which we are already familiar. For example, the behavior inherent in the phrase “teacher warmth toward children” may be conveyed by describing the teachers as friendly, intimate, or affectionate with her children. At this level, the behavior may be described without being observed or measured but simply in terms of related or associated concepts. A teacher’s clarity of presentation, variety of style, enthusiasm of manner, and organization of content are typical of behaviors described at this most general level. Because behaviors like these are described in such general terms, they must be tied to specific variables and competencies in order to be useful to the evaluation process. Variables and competencies, then, are derived from behaviors.

The word *variable* refers to the terms in which a particular behavior is to be observed and recorded. A variable specifies behavior by stating explicitly the way in which the behavior is to be measured. Variables redefine behaviors in terms of the operations that are necessary to observe and to measure them. These operations express the behavioral concept in the form of a measurement, which represents the level of differentiation at which the particular behavior can be reliably observed and distinguished from other behaviors.

Just as general behavioral concepts are used to derive variables, variables are used to determine the next level of behavioral description. *Competencies*, like variables, are characterized by a metric or scale. However, unlike variables, competencies include the specification of a desired quantity of behavior, which is referenced in the metric. Competencies identify a single level of proficiency, or a range of levels, determined through theoretical or empirical processes, at which a program participant should perform. Unlike variables, competencies are either attainable or not attainable. Hence, it is the level of proficiency which is critical, not—as in the case of variables—simply the separation and differentiation of various degrees of behavior. The process of deriving competencies from behaviors and variables is depicted in Figure 6.

In examining the decomposed model of a program, it is important to note whether the outputs are expressed in terms of behaviors, variables, or competencies, and whether or not they can be quantified at the competency level.

In the hierarchical decomposition of a program, the evaluator notes not only the conceptual precision with

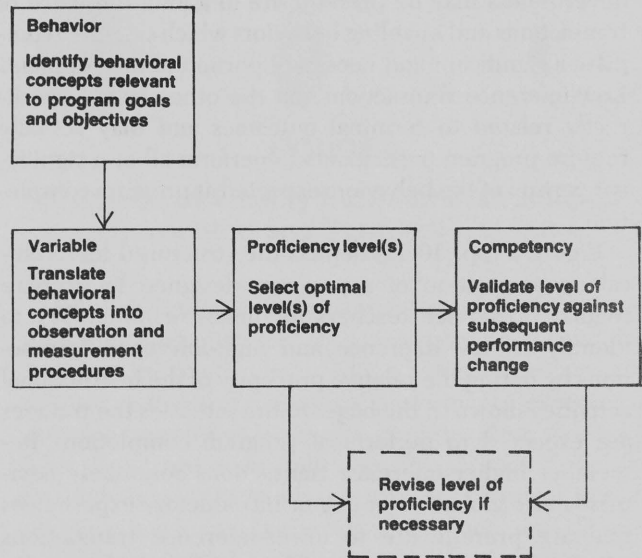


FIGURE 6

The Developmental Task of Deriving Competencies

which outputs can be translated into competencies, but also changes in output descriptions across levels of the decomposition—from knowledge competencies, which specify cognitive understandings, to performance competencies, which specify skills and processes. As implied earlier, high-inference transactions are commonly linked with knowledge outputs and low-inference transactions with performance outputs. Knowledge competencies often are the legitimate goals of high-inference transactions consisting of early program experiences; performance competencies, on the other hand, are the legitimate goals of low-inference transactions involving real-life tasks that will be encountered at program completion. It is performance competencies, therefore, that become the basis for summative judgments about the program’s effectiveness.

Outputs identified on the decomposition model are examined for their “quantifiability,” preferably at the competency level. In addition, the progression of outcomes from general to detailed levels of decomposition is examined for an increasing frequency of outputs that are expressed in terms of the participant’s competence on tasks expected at program completion.

DECOMPOSITION AND EVALUATION AS A TEAM EFFORT

The decomposition becomes a working document for project personnel to use in discussing the program. It by no means is intended to be impervious to change and critical assessment but, on the contrary, is meant to serve as an initial definition of the program and a vehicle by which to reconcile differing viewpoints held by members of the staff. Upon completion of the design phase, members of the team meet and each works through details of the decomposed model, usually prepared by the evaluator but reviewed by team members during development.

It is commonplace to learn at such meetings that each member of the team has a slightly different interpretation of the program. These differences often persist through the entire development phase and into program evaluation, complicating implementation and evaluation decisions. One purpose of the decomposed model is to identify and correct misconceptions among team members *before* development begins and to resolve inconsistencies and clarify ambiguities which may remain after planning and design are completed. The decomposition at this stage has four distinct effects:

1. Because its development is a team effort, it forces staff to use a common vocabulary and mode of expression in describing the program.
2. It exposes differing and sometimes extraordinary viewpoints of the program. It is not uncommon—and is in fact, healthy—at this stage to have various team members develop competing decomposition models from which to select a final version.
3. The decomposed model of the program serves as a framework in which to identify mismatches between transactions and intended outcomes (i.e., inferential leaps overlooked in the planning and design phase). Here, logical contingencies between transactions and enabling outcomes, and between enabling outcomes and terminal outcomes are a prime consideration.
4. The decomposed model serves as a framework for examining the nature of the outcomes intended. Outcomes that are stated as unoperationalizable behavior are replaced by more conceptually concise and quantifiable outcomes and, if possible, expressed as competencies to be exhibited by program participants. Also, the sequence of outcomes is closely examined to assure that those stated as knowledge competencies at general levels of the decomposed model are ultimately

transformed into performance competencies at detailed levels of the decomposition.

AN EVALUATION MODEL

After revisions in the program structure are made from the initial decomposition, the empirical work of the evaluator begins. This work entails the selection of transactions and groups of transactions to be evaluated and rests heavily on the structured decomposition of the program done during the planning and design phases. A model of the evaluator's task is presented in Figure 7.

The evaluator's work can be represented in six stages. The first stage involves reviewing program proposals and related documents and interviewing planners, designers, and developers about the program's objectives and purposes. From these data the evaluator, in cooperation with planners, designers, and developers, creates a structured decomposition model of the intended program—a model which is continually revised to increase the clarity of program concepts, to eliminate "inferential leaps," and to resolve differences in viewpoint which may exist among members of the development team.

The second stage is the decomposition itself. This stage becomes the foundation for all subsequent activities of the evaluator. Until the structured decomposition model of the program is completed, the evaluator's work is mostly qualitative and nonempirical in nature. However, completion of the decomposition model is a cue to the evaluator to begin the quantitative, empirical process of assessing the intended impact of the program and its components. The decomposition serves as a reference for the evaluator as he or she begins synthesizing "evaluative dimensions" which are gleaned from the individual diagrams (modules) of the decomposition model.

The evaluator arrives at these evaluative dimensions by working through three distinct entities that characterize the decomposed model. The first and most general of these is referred to as a subsystem. Subsystems are referenced on the decomposition model as the first diagram after the initial single-box description of the program has been written. Subsystems represent the initial breakdown of the global program into its components. The subsystems of our training program in Figure 5, for example, are shown in diagram AO.

The second entity of a decomposed model is the "module," which represents all subsequent transaction groups and further analyses within each of the subsystems. Modules are easily discernible because they always represent further division of subsystems. Modules are simply successive levels of detail within a particular subsystem and

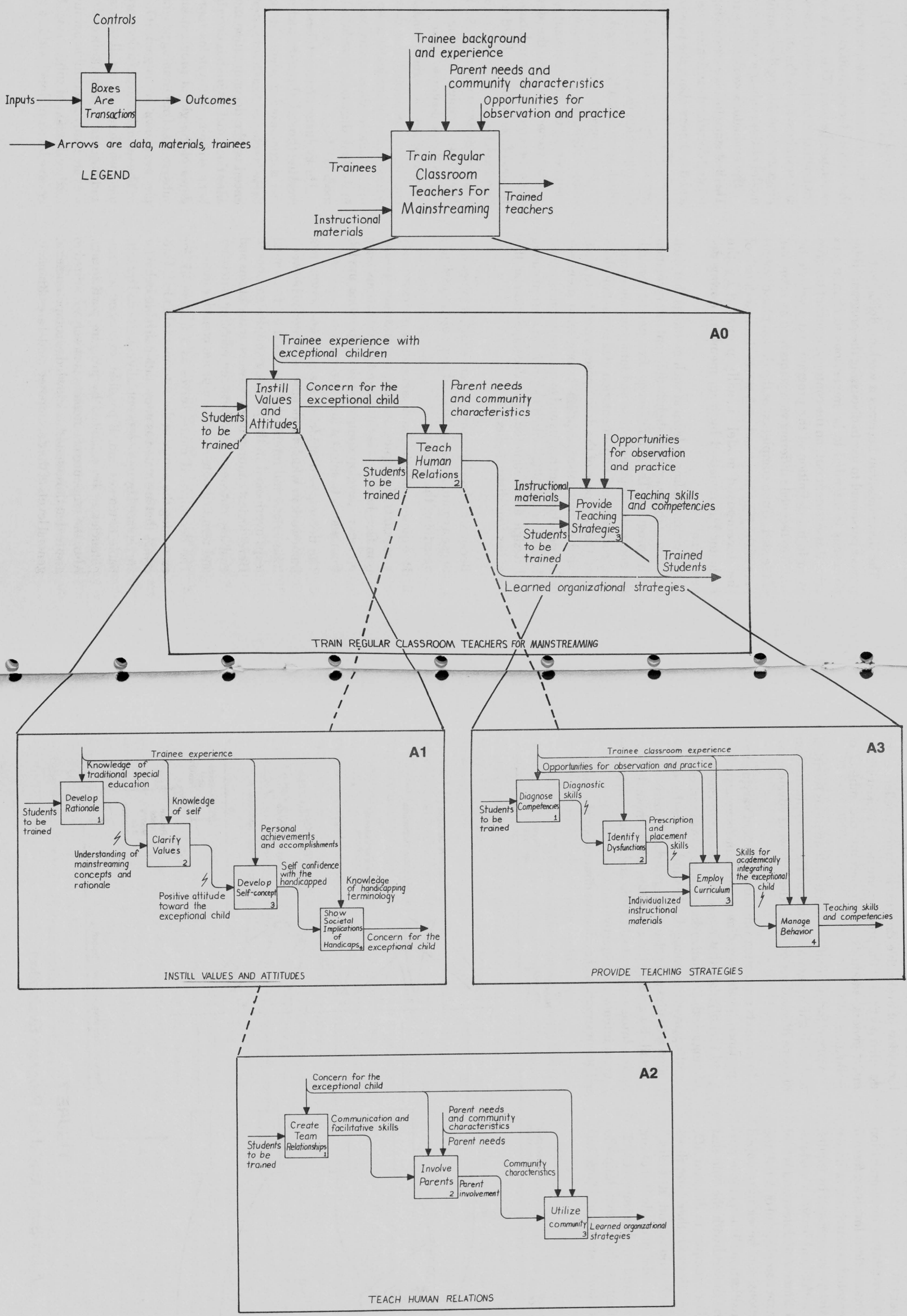


FIGURE 5
Hierarchical Decomposition of a Program to Train Regular Classroom Teachers for Mainstreaming
 (Using Structured Analysis and Design Technique) (SADT),[®] SofTech, Inc.)

are depicted as homogeneous groups of transactions. They always appear as a single diagram, or page, in the overall decomposition model.

The third entity within structured decomposition models is the transaction. Transactions involve a further level of detail and are represented as activities, or boxes, within modules. They are always interrelated by inputs, controls, and outputs both within and between modules at successive levels of detail. Since a module contains a set of homogeneous transactions at a single level of detail, the purposes of these transactions can be easily grouped under a single generic classification. This generic purpose, or *dimension*, may be defined with the title that defines the module itself. For example, in diagram AO, Figure 5, there are three modules and, therefore, three evaluative dimensions: "instill values and attitudes," "teach human relations," and "provide teaching strategies." They are represented by diagrams A1, A2, and A3, respectively. Evaluative dimensions help the evaluator to reduce the important concepts in a large and

complex program to a manageable number that capture the full flavor of the program. Thus, nothing is lost in the formation of evaluative dimensions since smaller, more detailed portions of the program are neatly tucked within more general modules which become the subject of evaluation.

A priori formulation of evaluative dimensions is critical to meaningful program evaluation. The conceptualization of these dimensions provides the following advantages:

- In projects containing voluminous data of varying importance, evaluative dimensions can focus activities and identify data that are most relevant to questions being asked. Evaluative dimensions provide criteria for setting priorities among the data and ensure that evaluation activities will not get "bogged-down" in irrelevant detail. By guiding the evaluation effort, these dimensions bring a conceptual handle and framework to program intents and objectives.
- The construction of evaluative dimensions also ensures

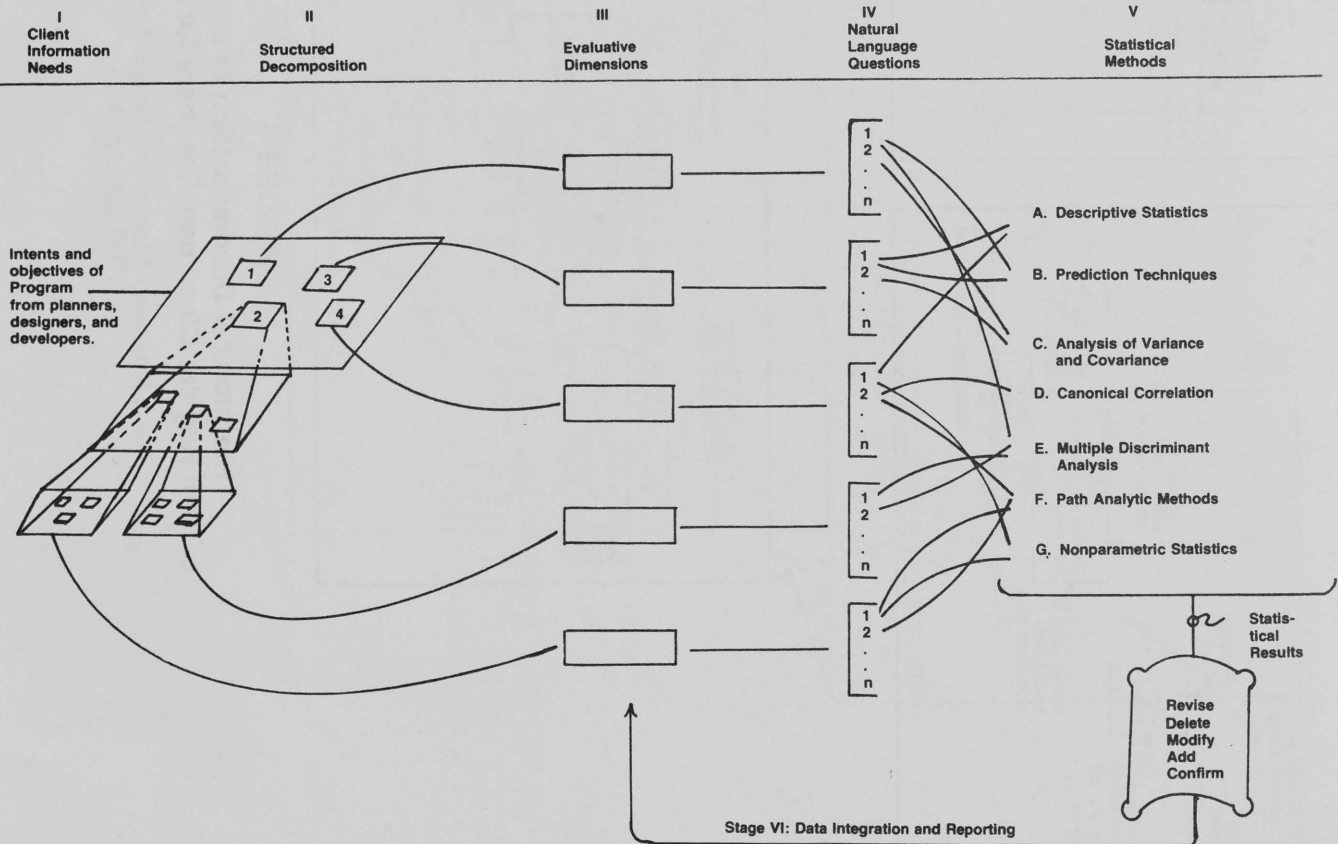


FIGURE 7

A Six-Stage Model for Program Evaluation

that subsequent evaluation activities will be congruent with the informational needs of the client organization. To be effective, evaluative dimensions must reflect what the client wants to know. Therefore, the development of evaluative dimensions is a critical component of the overall evaluation, because such dimensions link the client's priorities to the available data in a meaningful manner. In short, these dimensions ensure that the most relevant questions will be answered.

To be sure that these dimensions reflect appropriate informational needs and to obtain critical comments about his or her conceptualization of the evaluation effort, the evaluator submits the evaluative dimensions to the development team for perusal before collecting and analyzing data. This interaction can and often does yield conceptual insights about what sponsoring agencies want to know about the program, thus allowing the evaluator to become more "in tune" with sponsor intents and objectives for applying the evaluation findings.

Since evaluative dimensions necessarily group transactions into a single, more parsimonious configuration, evaluative dimensions often contain slices of the program structure which can be meaningfully evaluated. Thus, in Stage IV (Figure 7), the evaluator uses the evaluative dimensions to compose natural-language questions (expressed in everyday, common-sense terms) which can provide practical information to those who will use the evaluation results. Natural-language questions, therefore, are user-oriented, geared to those who must act on the results of the evaluation. They should have strong intuitive appeal to those who will implement and revise the program and a direct bearing on decisions that will be made in subsequent applications of the program.

After constructing and ordering natural-language questions from evaluative dimensions, the evaluator chooses in Stage V one or more statistical methods to answer these questions. The methods available range from descriptive statistics, which might identify trends using simple graphs, to more sophisticated techniques, which might pinpoint causal relationships between specific transactions and intended outcomes or differences between alternative versions of the same transaction. The evaluator need not have a thorough understanding of the internal workings of many of the statistical tools he uses, but simply a knowledge of the types of questions to which they apply and the computer programs by which to execute them. Many supposedly sophisticated techniques employing advanced mathematics inaccessible to the evaluator a decade ago can now be easily

implemented with computer programs available to virtually every computer installation in the country.

The seven statistical methods indicated in Stage V are groupings or sets of procedures—as opposed to individual procedures—which can be applied to a wide variety of natural-language questions. As the connections between Stages IV and V indicate, more than one statistical technique can, and wherever possible should, be applied to each natural-language question to cross-validate the findings relevant to that question. This procedure guarantees convergent validation of any conclusions drawn and enhances the credibility of the evaluator's report.

- A. *Descriptive statistics.* This grouping includes measures of central tendency and variability (mean, standard deviation, estimates of variance, and various graphing techniques) which can reveal trends in the behavior of program participants. A question that might be addressed by this group of statistics is: Does the implementation of program materials vary by teachers?
- B. *Prediction techniques.* This grouping consists of single and multiple variable regression techniques. These are used to show the relationship between variables. For example, does a teacher's use of program materials in the classroom vary as a function of the amount of experience she has had in using similar materials (i. e., is degree of implementation predictable on the basis of length of experience with similar materials)?
- C. *Analysis of variance and covariance.* This grouping consists of single and multiple classification comparisons. These techniques permit statistical analyses of the interactions between any two or more variables. For example, is degree of implementation of program materials highest when the teacher is familiar with similar materials (first variable) and is teaching a small number of students (second variable), and lowest when these conditions are reversed?
- D. *Canonical correlation.* This technique is the generalization of multiple regression to any number of dependent and independent variables. It identifies common variance in any two sets of variables and is used to study the underlying relations between these variables. Its most common applications have been to input-output analysis and cost-benefit analysis. For example, input variables might be (1) the number of hours teachers are trained in using program materials, (2) the number of staff and support personnel required, and (3) the number of instructional materials used; and output variables might include various mea-

asures of trainee performance, such as attitudes and cognitive understandings. Canonical correlations can identify various relationships between the two sets of variables and specify the contribution of each to the overall relationship.

- E. *Multiple discriminant analysis.* This technique is similar to regression analysis. It can be used to identify those variables that are most critical to participant performance. The technique allows maximum discrimination between groups of participants within the program. It can be effectively used to determine what variables best discriminate two groups of trainees on variables such as attitudes toward, knowledge of, and ability to execute specific teaching strategies. Statistical solutions might indicate, for example, that personality, prior training, attitude, and experience account for group differences.
- F. *Path analytic methods.* These techniques are used to hypothesize and test relationships among selected variables. They are applied primarily to determine causal relationships among variables. For example, it might be hypothesized that the extent to which a trainee implements specific teaching strategies is dictated by previous experience in the classroom, and to a lesser degree by attitudes and cognitive understandings. Path analysis indicates whether relationships between variables and outcomes are causal or spurious.
- G. *Nonparametric statistics.* This grouping includes techniques such as chi square (χ^2) and the sign test, which are employed when the data base fails to meet the assumptions required by the parametric methods described above.

The five foregoing stages tie available data to informational needs in a logically consistent manner. This model maximizes the information yield of an evaluative study, since each stage is built upon the preceding stage and all partners in the development process participate in formulating the evaluative dimensions and natural-language questions from which the data collection is planned. Involvement of the development team at various stages of program implementation ensures that evaluation activities will address relevant issues and provide additional data that may have been overlooked in earlier formulations of the program.

In Stage VI, the evaluator reports conclusions about each evaluative dimension based on results of the statistical procedures. As noted in Figure 7, conclusions are posed in terms of the original evaluative dimensions,

thereby giving statistical results continuity and an intuitive, common-sense appeal. The report is organized according to the evaluative dimensions (major headings) and the natural-language questions (side headings).

The evaluator's final task is to make recommendations to the program development team concerning the efficiency of various program modules. The answers to natural-language questions often have implications for specific transactions within modules. Consequently, evaluator recommendations are made at the transaction level whenever data permit. These recommendations, which are directed to the development team, reference specific aspects of the program for which data have been collected from evaluative dimensions and natural-language questions. They generally advise program developers to *revise, delete, modify, add, and confirm* given modules and transactions. Developers then proceed with the program changes for which personnel and fiscal resources are available. Finally, the structured decomposition model is revised to reflect the changes that are made and to accurately communicate the program in final form to all those who have a stake in it.

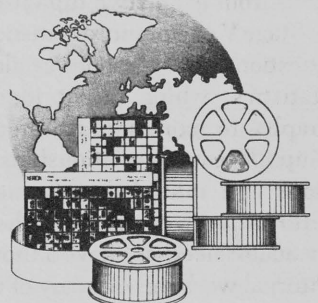
REFERENCES

- Bloom, B.S. (Ed.), *Taxonomy of educational objectives: The classification of educational goals*, Handbook 1, Cognitive domain. New York: David McKay & Co., 1972.
- Ross, D.T. Structured analysis (SA): A language for communicating ideas. *IEEE Transactions on Software Engineering*, January 1977, SE-3 (1), 16-34.
- Ross, D.T., & Schoman, K.E. Structured analysis for requirements definition. *IEEE Transactions on Software Engineering*, January 1977, SE-3 (1), 6-15.

Answers to Table 1: 4, 3, 2, 1.

**This
Publication
is Available in
MICROFORM**

FOR INFORMATION
WRITE:
Dept. FA.



University Microfilms International

300 North Zeeb Road
Ann Arbor, Mich. 48106
USA.

18 Bedford Row
London, WC1R 4EJ
England

CLASSROOM FORUM

*Edited by Melinda Young
Special Education Specialist
Vanderlyn Elementary School
DeKalb County, Georgia*

In my sixth grade classroom there is a child who seems to have mastered the skills prerequisite for writing, but his assignments are sloppy and inaccurate. He omits complete sentences and problems. His math papers are not headed nor are his problems numbered. Often he draws boxes around his math problems and lines between the different parts of his spelling assignment. Also, he makes careless errors in math, such as placing a number in the wrong column. What can I do to help this child to become more accurate and neater in his work?

The problems you have described affect many older children. Usually, these are children who experienced some visual-motor problems in the lower grades. In the upper grades the problem is seen as inaccuracy and a lack of organizational skills. Even at this age some visual-motor training may be appropriate, as well as help with organizational skills and accuracy. All training given should be as closely related to the real task as possible.

The following suggestions may be helpful. Some of these ideas, along with additional suggestions, may be found in Wallace & Kauffman (1973).

Visual-Motor Training

1. Encourage orderly development of movement patterns (e.g., working problems across the page in the book and across the worksheet, as well as working across his own paper). Use a card or piece of construction paper to help guide the movement.
2. Provide practice in organizing his paper visually. Place one row of problems on his paper and ask him to place the other problems under them. Use colored dots or other cues initially to facilitate placement of problems, if needed.
3. Use number cards to have the child reconstruct the

problems from the book. Draw lines on his paper to represent place value columns to facilitate copying numbers from the book and placing them in the correct column. Have him compare the problems. Praise him for correct placement of numbers. Later, place color-coded dots to represent the columns. Encourage the child to invent his own cues to help him organize his page visually.

Improving Accuracy

1. Allow the child to be the tutor to help other children with similar tasks.
2. Encourage the child to verbalize the math problem before writing the answer.
3. Allow the child to go to the free-time corner only when his work has reached a predetermined standard of accuracy.
4. Plot the percent correct on a graph and display it before the group. If accuracy is lacking in all subject areas, choose the area to display first in which accuracy is best.
5. Divide the class into teams. Compute the accuracy on the day's work for each team. Present the winning team with a badge or medal. Allow the winning team to choose the game to be played during a special 10 minutes in the room.

Neatness

1. Provide a good model. Include the heading, numbering, spacing, etc. in the model.
2. Provide structure. Write the problems in math and require the child to supply the answer. Write the sentences in English and have the child supply the correct word. Provide a ditto with part of the assignment previously written for the student. Gradually eliminate the structure.
3. Define standards of neatness. Specify requirements for leaving margins, skipping lines between parts, drawing no lines to divide parts, and using space to organize their papers. Set the standard slightly above the child's present work level.
4. Reward the child for successively more accurate attempts at the model. Let him be the "secretary" and write positive comments that the teacher dictates on other pupils' papers.
5. Select only one or two aspects of neatness on which to work at any given time. As the target areas are im-

proved, more areas may be added.

6. Use color codes to cue the child where to start and stop working as well as provide other needed visual cues.

Defining criteria and providing a good model often help children who have problems organizing papers and completing work accurately. The structure provided should be gradually withdrawn until it matches what is usually given to children of the same age. You may be surprised when the papers from your entire class improve if you use the above suggestions to some extent with your whole class.

REFERENCE

- Wallace, G., & Kauffmann, J. M. *Teaching Children with Learning Problems*. Columbus: Charles E. Merrill Publishing Company, 1973.

Several of my fifth and sixth grade Learning Disabled students have struggled through two years of basic sight words, consonants, vowels and blends; they have become discouraged and don't feel that they are learning. I need a technique to give them some quick success experiences to spark a little enthusiasm and confidence.

One useful technique combines elements of the phonetic and linguistic methods. Choose one vowel sound—for example, the ai and ay spellings of the long a sound. Make a list of words which contain the spelling pattern. Linguistic readers are a readily available source of word lists based on the sound spelling patterns you wish to teach; more difficult multi-syllabic words incorporating the pattern may be included to offer challenge.

Duplicate the word list and distribute copies to the students. Read the list with the children and stress the vowel pattern as you help them to sound out the unfamiliar words. As part of a discussion of word meanings,

guide the group in using the new words in sentences; far from being a dull activity, this can produce some lively discussion among the students.

Send the word lists home with a note to parents suggesting that the children practice spelling a few words each night. Give the students blank cards to make flash cards for use at home.

Prepare worksheets to give the students practice in reading and writing the target words—for example, sentences to complete by adding a word from the list, classification tasks in which the students list words which are “Things to Do,” “Names of Things,” “Words That Describe,” etc.

Most fifth and sixth graders enjoy competitive games. These may be used for reinforcement if you are careful to see that every child is a frequent winner. The following game ideas make drills a bit less painful for students:

- A laminated race track game board, a set of dice, and markers may be useful for several games. Write words on the game board if the objective is word recognition. When the objective is spelling practice, use a blank game board and have the children proceed by spelling a given word, rolling the dice and moving a marker along the track.
- Review definitions by telling the students, “I’m thinking of a word which means _____”; the students try to guess the correct word.
- Divide the group into two teams and let them collect points by spelling words orally, writing them on the chalkboard, or reading them from flashcards.
- Prepare a “spill and spell” game with blank dice. Mark several with the vowel spelling you are working on, the others with single consonants or consonant blends.

Children who have used these techniques over a period of several months have shown a marked improvement in reading vocabulary and spelling achievement.

Our thanks to Ms. Kathy Pacetti, LD-EMH Resource Teacher, Pleasant Ridge Elementary School, Gastonia School System, Gastonia, North Carolina, for providing the above suggestions.