Contents

ARTICLES

Alternative Methods of Task Analysis
A Comparison of Three Techniques
Wellesley R. Foshay

A Functional Analysis of Task Analysis Procedures for Instructional Design
Patricia Kennedy, Timm Esque, and Jerry Novak

Instructional Analysis
The Missing Link Between Task Analysis and Objectives
Cynthia K. Hoffman and Karen L. Medsker

Current Trends in Task Analysis
The Integration of Task Analysis and Instructional Design
Charles M. Reigeluth

The Process of Task Analysis
Integrating Training's Multiple Methods
Kenneth E. Carlisle

DEPARTMENTS

Book Reviews, edited by Allison Rossett
Metaphors We Live By, by George Lakoff and Mark Johnson
Reviewed by Paul Harmon

Analyzing Instructional Content: A Guide to Instruction and Evaluation,
by Philip W. Tieman and Susan M. Markle
Reviewed by Wellesley R. Foshay

ERIC Reports on ID, edited by Barbara B. Minor
Alternative Methods of Task Analysis

A Comparison of Three Techniques.

Wellesley R. Foshay
Associate Director, Quality Assurance,
Control & Training
Advanced Systems, Inc.
1601 Totme Road
Elk Grove, IL 60007

Introduction
Task analysis has recently taken on new significance. Instructional developers have long been accustomed to performing a task analysis as part of the front-end work for a project. But recently the term has also gained currency among cognitive psychologists researching human information processing. They tend to use the term in its broadest sense, to refer to the entire analysis performed to specify what is to be learned, after the needs assessment and before the actual design of instruction.

With this broadened definition have come analytical techniques which go considerably beyond the objectives-based behavioral approach with which instructional developers are most familiar. The newer approaches analyze concepts or—most recently—cognitive processes.

These new approaches present a problem for the practicing developer: Should the new techniques be used instead of the established behaviorally based ones, or should an integration be attempted? We will address this question by comparing examples of the three basic approaches and then conclude with some tentative recommendations about when each basic approach is likely to be of greatest use.

Methods Used
To represent behaviorally based analysis, we will use a version of Gagne’s learning hierarchy analysis, but we will not employ recently suggested modifications of the technique (Briggs, 1977). Strictly speaking, the learning hierarchy is not inherently behavioral. However, it is perhaps the most widely used technique for determining the structure of a task and an optimal sequence of instruction for that task. The objectives upon which it is based are often derived from behavioral analysis. The more recent additions to Gagne’s technique (the five-part objective, analysis of underlying concepts, principles and skills, and information-processing analysis) are omitted here in order to keep the example more “purely” behavioral. These enhancements represent a significant “cognitivization” of the learning hierarchy approach, and thus make it less useful for comparing behaviorally based and cognitively based task analyses.

The concept-based technique we have chosen is concept hierarchy analysis (Tiemann & Markle, 1978; Reigeluth, Merrill, & Bunderson, 1978). Analysis of concepts has taken a number of different forms, ranging from Gilbert’s (1962) Mathetics and Thomas, Davies, Openshaw and Bird’s (1963) matrix analysis, through M.D. Merrill’s (1977) concept elaboration structures. The method of concept hierarchy analysis used here has achieved some currency because it is relatively simple to use, and it highlights the structural characteristics of a concept network in a way that is prescriptively useful.

The information-processing analysis method we have used is P. Merrill’s (1978) flowchart adaptation of Scandura’s (1976) directed graph technique. This technique attempts to specify with relative precision the cognitive operations involved in solving a particular class of problems. While use of some kind of algorithmic representation is common among information-processing cognitive psychologists, the specific cognitive operations and processing parameters incorporated in the analysis vary from one researcher to another depending on the specifics of the cognitive processing model used. These differences generally are not addressed here, in part because the analysis has not been carried to a fine enough level of detail to make the differences relevant.

The task analyzed by these three techniques is an example of complex problem solving: debugging a computer program. This problem was chosen in part because of its familiarity to many developers, and in part because of the belief that it is complex problem-solving tasks such as this that make the strengths and weaknesses of the three approaches most apparent.

Learning Hierarchy Analysis
Following standard practice, the terminal objective has been placed at the top of the hierarchy (see Figure 1). The hierarchy contains four vertical “branches,” corresponding to the skills involved in debugging a program under each of the four general conditions in which a program can fail:

1. failure to compile
2. failure to link-edit (if this is done as a separate step), after successful compilation
3. failure to run, after successful link-editing
4. failure to produce expected output at the end of a run (due either to incorrect program structure or incorrect input data)

Note that standard practice for construction of a skill hierarchy calls for branches to occur only when the skills required differ, not when the problem structure differs. In this hierarchy, there happens to be a close relationship between causes of failure and skills required for debugging, but the relationship is not perfect. For example, the two types of unexpected output are lumped together because the differences between them usually are not clear at the beginning of the debugging process. Similarly, at the top of the hierarchy, two different causes can require the skill of reading a dump (a listing of the contents of a computer’s working memory), so the hierarchy converges at that point.

Note: An earlier version of this paper was presented to The Association for Educational Communications and Technology, 4 May 1982.
At the very bottom of the hierarchy, there are two skills which are among the first applied in the debugging process: determining if there is a problem, and determining the class of the problem (failure to compile, failure to link-edit, failure to run, failue to run correctly). These are placed at the bottom of the hierarchy because they are simplest to learn, although they also happen to come first in the debugging process. This is typical of learning hierarchies: The order in which a task is performed may not necessarily conform to the sequence of instruction, particularly if an instructional strategy such as backward chaining is used.

To simplify the discussion, we will examine in detail only the approach taken in the branch of the hierarchy which deals with failure to produce expected output. The skills involved in this type of debugging are:
1. identifying errors when there is an error message pointing directly to the problem
2. identifying errors when there is an error message pointing to a problem only indirectly
3. using trace code when the error message provides only indirect hints, or when there is no error message at all (trace code usually includes extra print statements to print out intermediate values of variables such as loop counters and other state indicators)
4. reading and interpreting a dump

The progression of these skills is from simplest to most complex. Notice that the hierarchy indicates that the instructional sequence includes "bypassing" some of the higher-level skills, such as dump reading. This is because it is possible to debug most problems in this class without using the higher-level skills. For example, a programmer using trace code can debug many programs without having to read a dump.

As with the other analyses in this paper, this one has been completed at only a relatively gross level. If the analysis of each component skill were completed, conditions and criteria for each objective would be specified. Also, more detailed objectives would demonstrate that the lower-level skills also contain sub-skills which are components of higher-level skills. This would further justify construction of the relationships between the component skills which are shown in the hierarchy.

Concept Hierarchy Analysis
As with the learning hierarchy, the
concept hierarchy also is based on the four types of program error situations (see Figure 2). In Reigeluth, et al.'s (1978) terms, this diagram is a "kinds" taxonomy which specifies the four kinds of error conditions which the learner may encounter.

Again to simplify the discussion, only the branch of the hierarchy dealing with unexpected output will be examined. This branch is extended somewhat in Figure 3. Following the taxonomic approach, errors due to incorrect input and errors due to incorrect design are identified as the two kinds of errors resulting in unexpected output. Errors due to incorrect input are further broken down according to each type of incorrect input. If the analysis were complete, this list of types would contain more components. For each type, the analysis would also show the corresponding critical features of the output typically generated: error messages, job log, program output, and dump components.

The branch labeled "how programs process inputs" represents a departure from a pure "kinds" taxonomy. If it were completed, the branch would enumerate principles of program operation which would apply both to programs which operate properly and those which malfunction due to incorrect input. By including such principles, it would be possible to teach the learner to apply them to predict probable outputs under each of the kinds of input error conditions listed in the "kinds" branch of the hierarchy. This, in turn, would make it unnecessary to teach directly the critical features of output generated by each kind of error condition. Instead, it would be necessary to provide practice in applying the principles to predict output generated by each error condition, using an appropriate set of positive and negative examples of outputs. Including principles in a "kinds" hierarchy in this way is unusual, and conventional concept analysis probably would not include it. It has been done here to illustrate one way of simplifying a concept hierarchy by emphasis on principles as well as concepts. It is appropriate to note that use of principles in this way may reflect the influence of an information-processing approach to the analysis.

Information Processing Analysis
The information processing analysis is represented in the form of a flowchart (Figures 4 and 5), following P. Merrill's (1978) suggestion. Representation of cognitive processes by some form of algorithm is common among cognitive psychologists, although other methods also are used. The algorithms are constructed to be consistent with the analyst's model of cognitive processing. Consequently, the set of mental operations and parameters will vary from one analyst to another. At the risk of oversimplifying a complex area of study where consensus among researchers is still a scarce commodity, the following "typical" principles were used in constructing the analysis shown in Figures 4 and 5:

1. Each process shown in the algorithm should be a fairly generic cognitive operation. This analysis presumes operations such as: recalling from long-term memory, holding in short-term memory, perceiving (by each sensory channel), decoding, chunking, discriminating, searching, and scanning.

2. Whatever operations are included, close attention must be paid to the relationships among them. For example, if the model of cognitive processing used postulates a central processor which operates on the contents of short-term memory (STM), then the algorithm must assure the correct information is in STM before each transformation occurs.

3. The algorithm should stay within the limits of human information processors. This includes precautions such as: limit the load on STM to 4 to 7 "chunks," allow for decoding and encoding processes surrounding use of long-term memory, and allow for the relative processing requirements of the various parts of the system. These factors in turn might influence the "chunking strategy" used to accomplish a given sub-task.

Obviously, the algorithm shown in Figures 4 and 5 is not a complete application of these principles. Like the other two analyses, it has been carried only far enough to permit comparison. A fully worked example of information-processing analysis for a skill this complex is beyond the scope of this paper.

The focus of this analysis is on the cognitive processes used by an expert problem-solver. The central thesis of the analysis is that the problem-solver covertly models the program's operation and predicts the program's "normal" output. Discrepancies between this prediction and the actual program output lead to the recognition that a "bug" exists and that the debugging algorithm must be called in. The actual process of debugging then is portrayed as involving
modification of the learner's internal model of the program by adding hypothetical "bugs" until its predicted operation yields predicted output that matches the actual program output data. This is the general problem-solving model suggested by Braune (1982).

Note that early in the problem-solving process, there is a decision on type of error, as there was in the learning hierarchy analysis. However, the decision occurs early in this analysis because it is done first; in the learning hierarchy, it was included early because it was a simple skill to learn.

Once it has been decided that an error condition exists, the next step is to choose which branch of the debugging algorithm to follow. There is a difference between the branches in this algorithm and those in the two previous analyses. In this analysis, the branches describe different cognitive processes, rather than different problem situations. Thus, unexpected output and failure to run are on the same branch, because the cognitive processes used in debugging them are the same: only the data in short-term memory and from long-term memory change. This kind of simplification of the task analysis is typical of this technique.

To facilitate comparisons with other analyses, only the branch for debugging programs which fail to run or which generate unexpected output will be discussed in detail (see Figure 5).

Once a problem has been encountered, the next step in the debugging process should be to search for error messages. If any are found, they must be decoded. This may involve simple recall of the meaning of the message, or it may involve looking the message up in a reference manual and decoding

Figure 1: Gagne-style learning hierarchy: Debugging
that explanation. Once the meaning of the message has been determined, the next step is for the programmer to internally model the program (including the link-edit structure) by adding bugs to it which are likely to generate the observed error message. When a match occurs between error message(s) predicted from the modified internal program model and the actual error messages observed, then the hypothesis about the cause of the program bug is confirmed.

If no error messages are found, then the debugging process takes a somewhat different course. The actual output generated by the program must be compared in detail with the expected output generated by the programmer's internal model of the program, parts of which must be recalled for the purpose. Discrepancies must be noted and collated. These discrepancies then become the input for the process of revising the learner's internal model. Once the discrepancies have been accounted for, the hypothesis concerning the program bug is confirmed.

With either set of input data (error messages or output discrepancies), the success of the internal modeling process is determined by a number of factors. For example, the degree to which the error messages (or output discrepancies) point to the actual problem significantly regulates the complexity of the task.

Previous experience in modeling for the particular pattern of error messages (or output discrepancies) encountered is also important, because a pattern of cues pointing to a particular error need only be recalled, if it has been previously "solved." An understanding of the structure of the program being debugged has already been mentioned as relevant. In addition, the programmer must know general principles of how programs of that type behave when they function—and when they malfunction. Finally, general problem-solving skills dealing with hypothesis formation and arraying of data to test a hypothesis are also important.

It should be clear that ability to solve problems of this type is controlled by a combination of general problem-solving skills and relatively problem-specific knowledge and experience. This is consistent with research on complex problem solving in other areas, for example, Elstein, Shulman, and Sprafka, 1976. This research also indicates that problem-solvers frequently stop the process of hypothesis generation and testing before all discrepancy data are accounted for. Instead, there is a tendency to systematically ignore or discount data which do not confirm an hypothesis, particularly when a large number of discrepancies exist. The process of developing a hypothesis (by modifying the internal model of the program), predicting its output and comparing that to observed data, is very complex. If this analysis were complete, considerable attention would have to be paid to specifying this process in greater detail.

Comparisons of the Three Approaches

The three analysis approaches differ considerably in techniques for representing the task. The differences of greatest interest relate to type of information identified, macro-level instructional sequence, and micro-level instructional sequence. These differences are summarized in Figure 6.

Type of Information. The type of information made explicit by each technique is different.

The learning hierarchy makes explicit the components of the complete performance, and specifies conditions and criteria for each. Apologists for Gagne might point out that a learning hierarchy analysis could yield much of the same information as the other techniques. For example, it is possible to map concept acquisition tasks into a behavioral analysis by including sub-tasks typical of concept acquisition, using verbs such as "identify," "define," "classify," or "discriminate." Similarly, cognitive processes can be referred to by including tasks with verbs such as "solve," or "predict."

However, it seems unlikely that a conventional behavioral analysis would
identify underlying concept structures or cognitive processes with the precision provided by the other two analyses. When conventional condition-behavior-criterion objectives are used, there is nothing inherent in the learning hierarchy technique that would lead the task analyst to consider explicitly the concepts and cognitive operations involved. Thus, when doing a behaviorally based learning hierarchy analysis, it is possible to include behavioral objectives dealing with concept acquisition and cognitive operations, but it is unlikely that most task analysts would do this. Instead, the analysis would probably be of the sort done here, with a focus on identifying a set of sub-skills which combine more or less additively into complete skills. The descriptions of skills and sub-skills would probably be in strictly behavioral terms, without examination of cognitive underpinnings.

By contrast, the concept hierarchy analysis explicitly identifies the concepts and the relationships among them (note that this is not true of concept analyses which do not specify a concept structure diagram such as the hierarchy). However, conditions, criteria, observable behaviors, and cognitive skills are implicit, rather than being stated explicitly.

On the other hand, the information processing analysis focuses only on the cognitive skills, although relevant bodies of concept knowledge frequently are identified, but not structurally analyzed. Conditions for performance of the skill are identified very carefully, since they constitute the set of input data for the cognitive operations mapped. However, criteria of performance typically are not specified, except for the implication that successful completion of the algorithm is the desired outcome.

**Macro-Level Sequence.** Additional differences can be identified at the "macro" level of instruction sequencing, that is, from the student’s entry level to full attainment of the global objective. The learning hierarchy is the only one of the three methods of analysis which directly suggests an instructional sequence at this level. Indeed, the principal reason for constructing a learning hierarchy is to determine the instructional sequence, which is always from the bottom of the hierarchy to the top.

By contrast, concept hierarchies present no such clear picture. Some authors (Reigeluth, Merrill, Wilson & Spillner, 1980; Engelmann, 1980) argue for a top-down instructional sequence in all cases. However, there is not a universal consensus. Tiemann and Markle (1978), for example, make no such recommendation. In the program debugging example, a top-down sequence would be one alternative, but other sequences are also possible. For example, a "middle-out" sequence might be used, in which the designer would start by teaching debugging in one class of problems and then generalize progressively to other classes of problems, continuing with an overview of the taxonomy of problem situations. Conceivably, the sequence in which the concepts are presented might also be influenced by individual differences variables such as previous knowledge or cognitive style.

The information-processing analysis also presents no clear guidelines for instructional sequence. Very little research has been done on how problem-solving structures as complex as the debugging example are acquired. If we apply Scandura’s (1976) framework to this example, we probably would not attempt to teach debugging in procedural order, using either forward or backward chaining.

Scandura’s argument is that complex problem solving rests on acquisition of a series of insights about the structure of the problem space which can not be comprehended by the novice. In this ex-
ample, such insights probably surround the ability to predict program function and malfunction. Scandura's principles might suggest a carefully-structured sequence of debugging problems which grow in complexity. Each increment in complexity would be designed to cause the learner to acquire more sophisticated insights about how to predict the function and malfunction of programs with different structural characteristics. Then the procedure used by the novice would differ substantially from that used by the expert. Instructional tasks would take this into consideration.

"Micro" Level Sequence. None of the three methods of analysis directly represent prescriptions for design and sequencing of instruction for a single concept or sub-skill. However, strong prescriptive principles are commonly applied to the learning hierarchy and to the concept hierarchy analyses. Prescriptive principles for information-processing analyses are less well-defined, but the analysis does make possible some useful insights in design. For learning hierarchy analyses, the most familiar prescriptions are summarized in Gagne's Events of Instruction model (Gagne and Briggs, 1979), although practically all of the literature on instructional design could be applied to a task analysis done in this framework.

The concept hierarchy also is associated with strong prescriptive principles. At least three procedures for concept teaching have been published (Tiemann & Markle, 1978; Merrill, Richards, Schmidt, & Wood, 1977; Engelmann, 1980). All three procedures provide highly specific guidelines for construction and use of concept definitions, sequences of positive and negative examples, and testing sequences. The level of precision in these procedures is much greater than those derived from typical behavioral analyses.

Prescriptive use of information processing analysis is not far advanced. Development projects using this tech-
nique are becoming increasingly common, but a set of coherent design principles which are grounded in research has yet to emerge. This does not mean that information processing analysis has no prescriptive utility. The algorithm derived from the analysis may be directly presented to the student in some form. Furthermore, the analysis appears to have great potential in providing detailed guidelines for design of test items, instructional problems and dialogues. In particular, the ability to make inferences about the state of the student’s learning from errors committed, is attractive (Scandura, 1976).

Conclusions

The three techniques chosen for this comparison are representative of three major approaches to task analysis: behaviorally based, concept-based, and cognitive process-based. It should not be concluded that the three analysis approaches represent incompatible extremes of a theoretical or ideological continuum. Instead, this paper has demonstrated that they differ only in what is made explicit and what is left implicit in the task being analyzed. If this is the case, the techniques associated with each basic approach are probably better viewed as variety of “lanes,” each with a different kind of “filter” through which to view the task. The “lane” and “filter” the developer chooses will depend on the nature of the subject matter and the effect desired.

The techniques vary in the precision with which they can be applied. However, all the methods involve considerable intuitive judgment on the part of the analyst, particularly in projects involving real instructional problems. Practical application of the techniques must, therefore, constantly raise the question of the cost of the analysis versus prescriptive benefits gained. The outcome of this consideration will vary considerably by situation, and it further precludes universal application of any one technique.

Recommendations

The example analyses presented here may be representative of a number of instructional situations involving troubleshooting complex systems, such as electronic devices, or living organisms. Such troubleshooting is one general class of complex problem-solving. It may be that in situations involving complex problem-solving the differences between the techniques become clearer. In any case, the examples presented here lead to four tentative recommendations for use of the techniques. Such recommendations can only be validated, of course, by considerable experience with each technique.

1. Use learning hierarchy analysis if “macro”-level sequencing and measurement problems are paramount, and supplementary analyses will suffice for identifying underlying concepts, principles and skills.

This recommendation is based on the strength of the learning hierarchy as a tool for sequencing instruction, and the strength of the instructional objective as a means of identifying conditions and criteria.

2. Use concept analysis if discriminations and relations among concepts are the most difficult part of problem-solving, but the cognitive processes involved are fairly simple.

This recommendation is based on the strength of concept hierarchies as a means of identifying concepts and concept/example sets for teaching relational structures among concepts.

3. Use concept hierarchy analysis if specification of conditions and criteria of performance is impractical.

In many situations, a behavioral analysis requires strong assumptions about conditions and criteria, because the analyst cannot predict specifically how learners will use the knowledge, or because the range of application situations is too broad or complex to serve as a useful guideline for task analysis. In these cases, concept hierarchy analysis may be a more suitable way of achieving precision in the analysis.

For example, in a recent analysis of diagnostic tasks in a medical subspecial-
ty (Foshy & Hatch, Note 1), a concept hierarchy provided the potential for elegant and precise prescriptions for instructional design while bypassing the problem of developing objectives for the entire range of possible diagnostic problems present in the subspecialty.

4. Use information-processing analysis if cognitive operations involved in problem-solving are complex, especially if detailed design of feedback or error analysis is a design goal.

This recommendation is based on the strength of information-processing analysis as a means of determining the details of the problem-solving process. It has particular importance for designers of simulations, games, and computer-based instruction. At this time there is, however, little consensus on guidelines for taking an information-processing analysis to a finer level of detail than that shown here. For the instructional developer, this means that information-processing analysis is probably more intuitive than would be desirable.

Of course, the techniques may be used in combination, depending on the analyst's judgment of the instructional problem. Given the state of the art, practitioners who work beyond the laboratory cannot afford the luxury of an ideological commitment to either behavioral or cognitive analysis. When dealing with complex instructional problems, it is probably best to retain an eclectic orientation, and learn to combine specific analytic techniques based upon the information generated. For this reason, case studies of such applications are needed if guidelines are to be derived for using the techniques together.

Reference Note


References


Merrill, M. "Concept Analysis via Concept Elaboration Theory." Journal of Instructional Development, 1.1, Fall, 1977, 10-12.


A Functional Analysis of Task Analysis Procedures For Instructional Design

Patricia Kennedy, Associate Professor
Timm Esque, Graduate Student
Jerry Novak, Graduate Student

Department of Educational Technology
Arizona State University
Tempe, AZ 85287

Abstract. Representative examples of task analysis procedures were examined for common components, methods, and terminology. Resulting generic components were categorized into two phases of task analysis: task description and instructional analysis. Task description included the components of task inventory, ordering, and refinement. Instructional analysis was comprised of the specification of needs, goals, objectives, learning hierarchy, learning taxonomy, training considerations, and product development specifications.

Identification of these components permitted the formulation of a generalized model of task analysis. The model contained a consensus of procedures comprising educational and industrial task analysis applications.

Task analysis is a procedure for identifying different kinds of performances which are outcomes of learning and sequencing structural units in the design of training or instruction. The roots of the concept of task analysis can be traced to contributions made by two researchers: Robert Miller (1953a, 1953b) in connection with military training research, and Robert Gagne (1968) who related task analysis to the design of instruction.

While serving as a generic term covering various types of “front end” analysis, task analysis contributes to research from both a theoretical and practical base (Dick, 1981).

As a bridge among theories of knowledge, instruction, and learning, task analysis serves three roles: as a prescription of the prerequisites and conditions under which optimal performance may occur or even what optimal performance is (a theory of instruction); as a description of the behaviors and processes through which performance may be efficiently achieved (a theory of learning); and as a means by which basic questions concerning the relevance and utility of performance may be explored (a theory of knowledge) (Davies, 1973).

To many, the accepted procedure of task analysis involves specifying a task's terminal components and subsequently ordering its instructionally useful, prerequisite subtasks (Gibbons, 1977). This procedure differs somewhat with Gagne's pioneering work in using task analysis to design instruction optimally for learning (Gagne, 1965, 1974). Gagne recommends that the learning task must be broken down into behavior capabilities that are not themselves the task, but are contributors to the performance of the task. The contributors are classified into types for identifying different optimal conditions for their learning. In practice, however, there are many ways of determining the hierarchical or linear relationship among the constituent elements and identifying the learning processes associated with each (Gibbons, 1975). Despite a long history of successful applications of task analysis, users lack agreement regarding terminology and methodology (Duncan, 1972). In fact, Duncan calls task analysis an art. Identification of the common components of task analysis procedures would be the first step in standardizing terminology and methodology.

In this paper, selected task analysis procedures are analyzed and contrasted after the isolation of key concepts and common procedural components. The identification of “common denominators” across the representative task analysis procedures permits the determination of the structural relationship between procedural components and formulation of a generalized model of task analysis with standardized terminology.

Methods

Procedure

An exhaustive review was conducted of all articles dealing with theory or application of task analysis published between 1979 and 1982 in National Technical Information Service, Psychological Abstracts, and ERIC. With additional articles identified in bibliographies, 415 articles in the public domain were reviewed. Proprietary models were not included.

Of the 415 articles, about two-thirds were immediately discarded. They were not useful for our purpose, because their abstracts did not contain a reference to discernible procedures for conducting a task analysis. Many of these were project-specific technical reports. Of the remaining 149 abstracts, availability of the articles was limited to 52. These articles were helpful for generating references to older articles describing task analysis procedures. The topic of the available references was varied widely, spanning learning theory (Wildman & Burton, 1961; Fleishman, 1978), developmental theory (Spada, 1980; Seigler, 1980; Swinton, 1977), diagnostic prescriptive special education applications (Arier and Jenkins, 1979; Burton, 1976; Ewing & Brecht, 1977), and job analysis training applications.
(Cornelius, Carron & Collins, 1979; Arvey & Moosholder, 1977; Prien & Rosen, 1971). From the 52 articles and their references, ten representative articles containing fully-developed methods were retained for further examination.

From this sample, the components of task analysis were listed in the terminology and order presented in the article. A Q-sort procedure was used to group the descriptions of components by similarity of process content. A generic term and operational definition were then applied to cover each group of discrete processes. The rationale for these ordering and labeling procedures was to derive our basis of comparison from the models themselves as opposed to comparing the models to a predetermined schema as was done elsewhere (Andrews & Goodson, 1980).

Results

Task Analysis Methodologies

A matrix was prepared of the ten task analysis methods cross-referenced against common components that were present in one or more of the methods (see Table 1). The term the author used to name the components was reported as well as the order in which the component was addressed in each method. This general model of task analysis contained all of the nonredundant components of the procedures we examined: task inventory, ordering, refinement, needs, goals, objectives, learning hierarchy, learning taxonomy, training considerations, and product development specifications. A discussion of each method is presented next, followed by a description of the common components of task analysis listed in Table 1.

In the list of authors in Table 1, Gagne's name is conspicuous by its absence. Most of the components of task analysis were first described by Gagne in his explanation of learning hierarchies and the relation of task analysis to content analysis (Gagne, 1962, 1965, 1974, 1977). However, a review of Gagne's extensive work did not reveal a step by step operational methodology of task analysis for instructional designers.

Of the selected task analysis procedures, two were applied to educational tasks. Both Resnick et al. (1973) and Merrill (1973) began a basic level of educational task description. Merrill proposed a detailed eleven-step procedure starting at the level of operationally defined concepts and relational operators. In addition, he addressed current issues in the theory of task analysis such as the inclusion of horizontal hierarchy analysis while evaluating prerequisites. With Merrill's method of analysis, the sequence of operations in a task is flowcharted with decision points or branches noted. This information processing approach employs the techniques of observation of overt behavior and "thinking aloud" of covert behavior. Unique is Merrill's position that neither hierarchical analysis nor information processing analysis is adequate for all skills. Both forms of analysis may be used to supplement each other by identifying instructionally important behaviors that would be missed by either approach alone. Proposed in more recent work by Merrill and his colleagues (Reigeluth, Merrill, Wilson, & Spiller, 1980) is an elaboration model which follows a general-to-detailed pattern of sequencing, as opposed to the hierarchically based sequences derived from a Gagne type of task analysis.

Resnick applied task analysis in the development of an introductory mathematics curriculum. Resnick's method of analysis resulted in a task hierarchy containing the terminal task on top, sequentially related sub-tasks next, and a hierarchy of learning tasks below each subtask. Resnick's "chain of component behaviors comprising a skillful performance" was a synthesis of Gagne's hierarchical analysis. Merrill's information processing analysis, and Scardura's directed graphs. Resnick's method both sequenced tasks and prescribed teaching strategies. Resnick agreed with Gagne and Merrill that verbal knowledge, as described by Gagne, was not amenable to this type of analysis.

We found ten nonredundant components: task inventory, ordering, refinement, needs, goals, objectives, learning hierarchy, learning taxonomy, training considerations, and product development specifications.

Many of the task analysis applications reviewed were industrial. Gard (1972) and Gilbert (1972, 1978) presented models of task analysis applied to industrial training tasks. Gard reporting on an application for training systems designers, described a four-step procedure for task analysis: Clarify tasks as activities, inputs, or outputs; organize a hierarchy; assign a taxonomy value; and achieve expert consensus of the analysis. Gard's method of analysis, which can be considered a gross level of information processing, was similar to Merrill's. Output of gross analysis methods served as input to other methods of analysis. Gilbert's rationale for his task analysis procedures was based on cost effectiveness considerations in achieving human competence, not psychological theory (Gibbons, 1977). He focused on reducing training costs by isolating deficiencies with a knowledge and learning taxonomy matrix when suggesting training in the deficient areas. The estimated payoff of reducing mismatches between actual and desired performance was used to set priorities for training.

The products of some of the industrial applications were highly detailed. Miller (1962) and Duncan (1973) formulated their task analysis procedures from applications to highly procedural jobs. For Miller, task analysis was "a process whose results provide data about human functions, which in turn are used to determine the character of the system and its components." The main difference between Miller and Duncan was the use of learning taxonomies. Miller prompted the use of a learning taxonomy to prescribe instructional methods while Duncan maintained that task analysis was prescriptive only for determining training sequences and not for specifying training methods. Another detailed product was Martin and Brodli's (1972) task-based curriculum for hospital corpsmen. The tasks were categorized by means of syntactic clustering, a method of classifying
Table 1

Methods of Task Analysis Cross-Referenced by Components
(The order each author addressed components is indicated by numbers.)

<table>
<thead>
<tr>
<th>Task Inventory</th>
<th>Ordering</th>
<th>Refinement</th>
<th>Needs</th>
<th>Goals</th>
<th>Objectives</th>
<th>Learning Hierarchy</th>
<th>Learning Taxonomy</th>
<th>Training Considerations</th>
<th>Instructional Specifications Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Herschbach</td>
<td>#6</td>
<td>#7</td>
<td>#2, 3, 4, 5</td>
<td>#1</td>
<td>#10, 3, 9</td>
<td>#2</td>
<td>#11</td>
<td>#4</td>
<td>#11</td>
</tr>
<tr>
<td>Gilbert</td>
<td></td>
<td></td>
<td>#1d, e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Construct a knowledge map which organizes method and sequence</td>
</tr>
<tr>
<td>Gard</td>
<td></td>
<td></td>
<td>#3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merrill</td>
<td>#1</td>
<td>#2</td>
<td>#4</td>
<td>#5</td>
<td>#6</td>
<td>#7</td>
<td>#9</td>
<td>#10</td>
<td>#11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>#3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Classify behaviors associated with defined problems as rule-gaging or rule-finding</td>
</tr>
<tr>
<td>Miller</td>
<td>#1</td>
<td>#2</td>
<td>#3</td>
<td></td>
<td>#6</td>
<td></td>
<td></td>
<td>#7</td>
<td>#3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>#3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Consideration of information processing components</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>#4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#8</td>
<td>#1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>#3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Methods derived from AB Sequence derived from chronological task description of behavioral clusters</td>
</tr>
<tr>
<td>Task Inventory</td>
<td>Ordering</td>
<td>Refinement</td>
<td>Needs</td>
<td>Goals</td>
<td>Objectives</td>
<td>Learning Hierarchy</td>
<td>Learning Taxonomy</td>
<td>Training Considerations</td>
<td>Instructional Specifications Development</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>------------</td>
<td>-------</td>
<td>-------</td>
<td>------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>-----------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Hannum</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Identify learning outcomes</td>
<td></td>
<td></td>
<td></td>
<td>#5 Define training sequence as defined by prerequisite analysis</td>
</tr>
<tr>
<td>Martin/Brodil</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Identify behavioral contributors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duncan</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Task statement analysis (training need)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Categorical needs analysis (patient needs)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gregory</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Hierarchize major operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Define the plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resnick</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Develop a task activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Identify and diagnose training problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Need assessed by testing objectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specify concept components in behavioral terms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mitigate terminal and enabling objectives by prerequisite analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Exit related action models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Roll action models to derive the node</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#6, 7 Hypothesize, test and evaluate methods of instruction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Two different sets of procedures for conducting task analysis seem to be needed: one for educational and one for industrial designers.

Tasks were described in terms of "action models" which incorporated mental events into the task structure. Full utilization of this model, still in the early development phases would entail the use of user-friendly computers for gathering task data from experts and novices. The computer program would delineate an action model for training analysis from the differences between experts and novices on answers to questions related to job performance. All of these applications of task analysis possessed what Gerlach, Reiser, and Breck (1977) call "generality." That is, each contained a description of task analysis procedures complete with enough detail to be applicable to a class of problems as opposed to a single problem. Instructional developers could follow any one of the task analysis procedures discussed here. It was felt that generality was an important selection criteria in a review such as this one.

Common Components and Standardized Terminology

It seemed that the task analysis components could be divided into two phases: task description and instructional analysis. Description dealt with the inventorying, ordering and refining of task content. Analysis covered the use of task analysis for the analysis of instructional design specifications for a given task or series of tasks. In the latter phase the design procedures for systematic development were addressed by the specification of needs, goals, objectives, learning hierarchy, learning taxonomy, training considerations, and product development specifications. A discussion of the common components follows.

Most authors addressed the three task description components, although the order and mass they gave to these components varied. Task inventory involved a progressive redescription of task elements from global to detailed. Ordering referred to the arrangement of task elements according to their content or performance relationships. Refinement contrast, other authors, generally those concerned with education rather than training, conducted needs analysis late in their procedure. For instance, Merrill analyzed needs in step eight of his eleven-step procedure. Both Merrill and Resnick suggested compiling a statement of needs from the information obtained after administering a pretest to identify student achievement deficits. The nature of the needs varied among training needs, learner needs, and recipient needs. For example, Martin and Brodt conducted a task statement analysis for training needs. Other authors such as Resnick, Gilbert, and Merrill identified learner needs. This distinction reflected a difference between examining a task and examining a learner's behavior for determining need.

Goals were defined as broad statements of instructional aims. A noticeable finding from the matrix was that six of the ten authors did not include defining goals as part of their task analysis procedure. However, authors who did not use goals usually specified objectives. The specification of objectives appeared to be the product of the task description phase while the goals were the product of the instructional analysis phase. When the needs analysis centered on the task, as with Gard's procedure where the difficulty and frequency of specific activities were rated, objectives were viewed as synonymous with the final product of the task description phase usually in hierarchy form. For example, Gard did not begin task analysis with task description. Rather, he began by classifying tasks as activities, inputs, or outputs. Then he devised a behavioral hierarchy by sequencing this task information. Thus, for every activity there was an input and output. In theory, objectives are derived from goals (Mager, 1972). In practice, as reflected in the matrix, this was not how things worked. Gilbert and Gard, for instance, did not do a task description. Their objectives were not derived from goals but came from behavioral statements. In task-based models, there was not a clear relationship between goals and objectives, as basic tenets of development would lead one to expect.

Objectives were always stated in behavioral terms whether they were generated from the task description phase or a priori. Only one author, Duncan, did not include objectives in his task analysis. He went directly from his task description to a training sequence which he empirically validated. His ap-
approach circumvented learning hierarchies and taxonomies.

The learning hierarchies component was defined as an arrangement of behavioral objectives according to prerequisite learning requirements. It is synonymous with prerequisite analysis in Gagne's terms. It was found that not many authors devised hierarchies. Resnick formed hierarchies by prerequisite analysis of terminal and enabling objectives. Gregory's interactive procedure started with posting a task. With an action model, he expanded this initial task into behavioral objectives. Simultaneously, he derived the prerequisite and subsequent tasks, thus generating a learning hierarchy.

Gagne (1977) suggested that the learning hierarchy serve as a main source for determining training sequences. What is interesting to note was that some authors, namely, Miller and Duncan, derived their training sequence from the hierarchical description of the task as opposed to a prerequisite behavioral analysis. Implied is a treatment of the task hierarchy resulting from the task description as synonymous with a learning hierarchy.

Learning taxonomies were defined as the classification of task-related behaviors within the parameters of pre-defined types of learning. Only four authors used learning taxonomies in their task analysis. Hannum followed Gagne's taxonomy. Information processing taxonomies were utilized by Miller and Merrill. Gilbert's taxonomy was what he called a knowledge progression going from theory to application in five steps without employing information processing terms. We concluded that the purpose of the learning taxonomy was for determining teaching methods, while the purpose of the learning hierarchy was for sequencing, so the two were not interchangeable in our model.

Training considerations were defined as an analysis of factors that constrained the scope of the instructional product. The educational applications of task analysis, Resnick's and Merrill's, ignored training considerations. Some industrial applications, such as Duncan's, included delimiting trainer responsibilities from those responsibilities that were rightly the organization's.

The final task analysis component, instructional specifications development, was defined as the prescription and sequencing of instructional methods. Authors varied in the techniques used for prescription and sequencing. Gilbert maintained that method and sequence were determined by the learning taxonomy. Hannum derived his sequence from a prerequisite analysis of subtasks. Miller utilized a chronological task description of behavioral clusters. In contrast with this group of authors were Duncan and Resnick who maintained that while sequence could be prescribed by the task description, instructional methods could only be hypothesized. That is, rather than using a taxonomy to prescribe given teaching techniques, they strongly suggested more freedom with regard to teaching methods as long as the results could be validated empirically.

Discussion
From the inspection of the rows, it can be concluded that no author's procedure was complete in addressing all components of task analysis. Also, from inspection of the columns, it can be concluded that no component was addressed by every author. Finally, the order in which the common components were addressed varies from author to author. Most, if not all, task analysis procedures were generated primarily for their idiosyncratic application. It is clear that the procedures for task analysis did not evolve systematically. Rather, different procedures proliferated in isolation, probably in response to the situation specific demands.

An exhaustive review was conducted of all articles dealing with theory or application of task analysis published between 1979 and 1982.

The most salient difference between approaches depended on whether the application was educational or industrial. People in industry approach task analysis differently from people in education, probably due to the nature of the tasks and an emphasis on cost-benefit analysis. The educational techniques focused on jobs which were then analyzed into tasks and subtasks.

Instead of breaking tasks into concepts, industrial developers divided jobs into sub-skills which were not concepts but were actions. In other words, educational applications worked at a cognitive level, while industrial training models worked on a directly observable level of behavior. Industrial applications of job analysis stopped just short of the cognitive analysis found in educational applications. Another difference between educational and industrial applications was in how needs were assessed. In educational applications such as Merrill's and Resnick's, needs were assessed by pretesting students or objectives. In industrial applications, training needs were identified before objectives were written.

In both industrial and educational models, specification of objectives did not routinely follow specification of goals. No author used goals only, but many authors used objectives only. Objectives were always stated in observable, measurable form. It may be that the specification of goals is passe.

There was confusion between task hierarchies and learning hierarchies. The task hierarchy is derived from the task description, while the learning hierarchy is the product of prerequisite analysis. They are not interchangeable. (Editor's Note: See article by Medsker in this issue.)

From the differences in the nature of the tasks analyzed by educational and industrial designers, it can be speculated that two different sets of procedures for conducting task analysis are needed. The designer would look at the nature of the task to be analyzed to determine if all of the procedures recommended in our general model of task analysis would have to be completed. Further research is necessary to determine the payoff for doing all of the steps. The results of this research would have implications for education and practice, should it prove necessary to prepare instructional designers in two methods of conducting task analysis. While the difficulties of researching the area of instructional design as a whole have been well articulated (Dick, 1981), and apply to task analysis, breaking task analysis into components and researching the components are feasible research activities.

Further research not only might be directed to validating the components of
the general model of task analysis, but also might well determine the most efficacious way of accomplishing each component. From the columns of the table, it can be seen that there are at least a couple of ways of accomplishing each component, and the most efficient manner has yet to be determined from an empirical comparison or from well-documented case studies.

In view of the diversity of procedures, terminology, and applications and paucity of empirical comparisons, it is not possible to recommend one best method of task analysis for all problems. Rather, we argue for standard terminology and further research, including assimilating additional models into our schema.

References


Instructional Analysis
The Missing Link Between Task Analysis and Objectives

Cynthia K. Hoffman
and
Karen L. Medsker

AT&T Bell Laboratories
6 Corporate Place
Piscataway, NJ 08854

Abstract. In the design of industrial instruction, it is important to maximize learning and transfer to job performance, while minimizing learning time. Both task analysis and instructional analysis are essential processes for achieving these goals. Task analysis is a tool for understanding and specifying the desired final performance or job. The product of task analysis may be a task list, flowchart, and/or other documentation which describes competent performance. Instructional analysis, as distinct from task analysis, identifies the type(s) of learning involved in acquiring a new performance capability and the structure of that learning in terms of component skills and their relationships. The product of the instructional analysis process is a learning map.

This paper describes the rationale for distinguishing between task analysis and instructional analysis, the contributions that both processes make to the design of instruction, and why one is dependent upon the other. The major functions included in the instructional analysis process are described and illustrated with examples. The benefits of using this approach to achieve effective and efficient instruction are summarized.

Background

Gagné (1974) noted that training designers often fail to distinguish between “job-task analysis” and “training task analysis” as early as 1956. This critical distinction is still ignored, or blurred, by many leading instructional design practitioners in the models they offer to the field, as evidenced by a close look at Gibbons’ review of content and task analysis methodologies. Some models (e.g., Davies, 1973) advocate the development of instructional objectives directly from the task analysis, thereby introducing dangers of excess and/or inadequate instruction. Other approaches, for example, Essef’ and Essef (1974), do prescribe separate processes for analysis of the job and analysis of the learning components but oversimplify the relationship between the two processes. More is involved than just rearranging the boxes from a task analysis flowchart to form a learning hierarchy or map. Still others, including Briggs and Wager (1991), merge the two processes into a single design phase. If the instructional analysis is not based on the results of a previously conducted task analysis, the designer cannot systematically ensure that the instruction will be job-relevant.

Failure to distinguish between the two types of analysis may stem partly from the frequent application of the term task analysis to both processes. We, therefore, prefer to reserve the term task analysis to mean the analysis of a job into its component tasks—what one must do to perform the job. The “job” refers to any performance; it is not limited to performance in one’s occupational area. The term instructional analysis is more descriptive of the second process, the analysis of a task into its learned components—what one must learn to be able to perform each task.

The model of instructional analysis discussed here is derived from Gagné and Briggs (1979) and Briggs and Wager (1981). It is based on Gagné’s taxonomy of learning outcomes and provides a structured and verifiable method for analyzing tasks into learned components. While other models exist for “deriving skills and knowledge” from task analyses, we find this approach most useful because it is based on relevant learning theory: it is efficient to use; and it is particularly appropriate to technical skills learning. There are also some significant payoffs in using this model for designing instruction.

Payoffs of Instructional Analysis

First, the instructional analysis process increases the probability of identifying all learning that is prerequisite to performing a task. Because intellectual skills are hierarchically related, for example, we expect prerequisite rule learning to be identified whenever a problem-solving task is the focus of the analysis. We know to search systematically for all essential rules and, in turn, their underlying concepts. Similarly, because of hierarchical and other supportive relationships within learning maps (the output of instructional analysis), instructional sequencing decisions are facilitated. Also, because types of learning are identified during instructional analysis, the foundation is laid for generating appropriate instructional strategies which correspond to each type of learning.

Another important payoff is the elimination of unnecessary instruction. If skill or knowledge items do not emerge during analysis of the desired goal behaviors, they are not necessary in the instruction. Thus, if “nice-to-know” items are included, it is with the knowledge that they are nonessential.

Finally, by doing instructional analysis as a separate step, the analyst can be optimally effective during task analysis by focusing on performance alone, without confusing it with how the performance is learned.

The Distinction Between Task Analysis and Instructional Analysis

Process and Products

Task Analysis. A task analyst asks the question, “How is this job or function (ideally) performed?” and documents the answer in a task list and/or flowchart. The task analysis process identifies all subtasks, information flow, inputs, and decisions required to perform a job. Figure 1 is the flowchart...
which resulted from a task analysis of the job of “conducting formative evaluation of an instructional product,” a job performed by most course developers.

The top row of boxes represents major functions in the process of conducting a formative evaluation. This is a high level of task analysis. The second and third rows are a further breakout, or list of subtasks, for one of the major tasks—conduct review of content accuracy.” Finally, a third level task analysis is presented under the subtask “consolidate feedback.” Three unordered activities occur here, whose output is input for “summarize feedback data.”

This is, of course, not the only way to conduct formative evaluation, but it represents a consensus among several instructional technologists at Bell Laboratories on how the job is best performed, based on observation and consultation with several practitioners in the field. No attention was given to learning during this task analysis; the focus was on how the job is best performed.

The flowchart (Figure 1) is an essential part of the instructional analysis process in which the tasks were analyzed in terms of their learning requirements.

**Instructional Analysis.** Instructional analysis requires a different mental set on the part of the analyst than does task analysis. The question asked is, “What must be learned in order for someone to perform this task?”

For an example of instructional analysis and its product, refer again to the job of conducting formative evaluation. Contrast the learning map (Figure 2), the product of instructional analysis, with the flowchart (Figure 1) of the task analysis. Specifically, note that in the task analysis, many subtasks emanate from one major task, “conduct review for content accuracy.” Why is the difference so great between the products of two analyses?

From a performance viewpoint, there are many steps involved in the task of conducting a content review. From a learning viewpoint, however, none of these steps, or subtasks, would constitute new learning. Members of the target population could, without additional instruction, “identify possible subject matter experts” or “make arrangements for a content review.” Since these behaviors are already in their repertoires. The only new learning that is required here is the concept “content review.” If the learners can demonstrate acquisition of this concept, they are, hypothetically, able to apply previously-learned skills and knowledge such as organizing, coordinating, and planning, to the task at hand—“conducting a content review.”

The flowchart resulting from the task analysis would not be used box by box, as a basis for developing training objectives, but perhaps as a job aid to prompt learners back on the job as they go through the steps of conducting a content review. It is the learning map, instead, that is the basis for developing training objectives.

Since specific tasks resulting from task analysis are often eliminated as candidates for training via instructional
analysis, training efficiency is often improved with this approach. Training can also be made more effective with this approach, since a focus on learning generic concepts and principles will facilitate a whole class of behaviors needed for a job and, therefore, enhance transfer of learning from the training situation to a variety of similar situations on the job. Transfer is less likely to occur when a task is taught simply by exercising its steps in the limited number of contexts available in the training setting.

Conducting Instructional Analysis

We have found the Gagne and Briggs (1979) and Briggs and Wager (1981) approaches to instructional analysis to be extremely useful in actual practice. The process described here is an extension of their technique, instructional curriculum mapping. Our approach differs in that we propose the use of a hybrid “type of learning” that can make the instructional analysis process more efficient.

Instructional analysis is not a linear process. It is iterative, as are all processes in systematic instructional design. For simplicity of explanation, it is described here as a linear process (see Figure 3).

Actually, the designer must move artfully back and forth between functions, integrating each of the function outputs with the other, to form the total learning map—the product of instructional analysis.

Function: Identify Types of Learning

The instructional analyst first attempts to determine what types of learning are required for trainees to acquire the skills necessary to perform the job tasks. This classification by type of learning is required in order to select the most appropriate analysis method(s).

Figure 4 summarizes—and briefly defines the five domains of learning identified by Gagne, lists designer verbs associated with each type of learning, and illustrates each with a sample task from a course in Structured Systems Analysis. For complete coverage of these types of learning, see Gagne (1977) or Gagne and Briggs (1979).

In the “conduct formative evaluation” example described above, concept learning was required instead of procedures learning, because members of the target population already knew how to do the separate procedures, but they could only be accomplished competently with acquisition of the underlying concept.

A Hybrid Type of Learning

—Complex Procedures

In performing instructional analysis for many of the training development projects at Bell Laboratories, we have found it useful to identify a sixth type of
learning which is often a combination of two or more of the other five types of learning and has an inherent sequence. We've labeled this hybrid complex procedure learning. Tasks which represent this type of learning have the following features:

- The learning of a sequence is required for successful performance.
- The major learning involved is the process itself and the inter-relationship of its components.
- The procedure is complex enough to require practice and feedback for shaping (i.e., a job aid alone would not be sufficient to assure mastery as it is in simple procedures).
- Although the components may be intellectual skills, they do not need to be analyzed as such because their subordinates are generic intellectual skills which support many terminal behaviors and have therefore already been learned by the target population.

We do not consider the number of steps and decisions as a relevant criterion in determining whether a procedure is simple or complex.

Note the sample analysis of a complex procedure, Figure 5, in which the subordinate intellectual skills of additional and subtraction could be assumed for most adult target populations and need not be analyzed into hierarchies.

Other examples of complex procedure learning include:

1. Conceptual Procedures or Methodologies
   - Use a "formative evaluation" model.
   - Apply a structured approach to analyze systems.
   - Analyze performance problems according to a particular model.
   - Manage a software project systematically.

2. Physical Procedures with Mental Processing
   - Changing a tire
   - Flying an airplane
   - Raising the sails on a sailboat

One might argue that a designer can identify these skills, their components, and their sequence just as effectively by searching for them in the context of Gagne and Briggs' five domains of learning. This may be true, but we have found that it is more efficient to conceptualize these types of tasks as complex procedures at the beginning of the instructional analysis process, first identifying the components of the procedure to be learned and then identifying any contributing intellectual skills which may need to be learned by the target population. In this way we avoid analyzing all tasks into hierarchies of intellectual skill learning when it is likely that only a portion of the skills will need to be taught as such.

Function: Procedural Analysis

The first step is to scrutinize the task analysis flowchart to distinguish complex procedures from simple procedures. Unlike complex procedures, simple procedures are those tasks that do not require new learning for the target population but can be performed with the use of a job aid. However, if the target population is to be able to apply the underlying concepts and rules that drive the simple procedure in a variety of non-training situations that they may encounter on the job, hierarchical analysis should be conducted to identify those concepts and rules for the learning map (see next section).

Once complex procedures have been identified, the analyst must review the task analysis to assure that all components of the complex procedure have been identified. Each component is then analyzed to determine if there are any intellectual skill prerequisites which must be learned by the target population. If so, they will be further analyzed during hierarchical analysis (see next section). If not, a pure complex procedure has been identified for the learning map (graphically displayed in Figure 6), and no further analysis is required.

This is a rare occurrence, however, so combination analysis is often required.

<table>
<thead>
<tr>
<th>Types of Learning</th>
<th>Definitions</th>
<th>Designer Verbs</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intellectual Skills</td>
<td>Demonstrating an ability to perform a whole class of behaviors, often using symbols, &quot;tinning hats.&quot;</td>
<td>Generate</td>
<td>Given prose narrative describing a system, generate a data flow diagram</td>
</tr>
<tr>
<td>Problem-Solving</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concepts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attitude</td>
<td>Acting on the basis of feelings, beliefs, or opinions (informed choices)</td>
<td>Choose</td>
<td>Choose to implement data flow diagrams on the job</td>
</tr>
<tr>
<td>Cognitive Strategy</td>
<td>Managing one's own thinking and learning processes</td>
<td>Originiate</td>
<td>Originiate a new design for a computer system</td>
</tr>
<tr>
<td>Verbal Information</td>
<td>Stating facts, names, ideas, generalizations, or processes that can be memorized, &quot;knowing what or that.&quot;</td>
<td>State</td>
<td>State the definition of data dictionary</td>
</tr>
<tr>
<td>Motor Skill</td>
<td>Executing a physical action with smoothness and timing.</td>
<td>Execute</td>
<td>Execute a backhand in tennis</td>
</tr>
</tbody>
</table>

*These are few motor skills required of the systems analyst.

Figure 4. Types of Learning (Adapted from Gagné and Briggs, 1979)

JOURNAL OF INSTRUCTIONAL DEVELOPMENT
Balance a Checkbook

This is a term borrowed from Dick and Carey (1978) which we use to describe the integration of complex procedures and supporting intellectual skills in the development of the learning map (graphically displayed in Figure 7). Actually incorporated into the new, higher level learning. This structure serves as a guide to designers in conducting hierarchical analysis of intellectual skills.

**Function: Hierarchical Analysis**

Any intellectual skills identified during procedural analysis should be thoroughly analyzed to determine any subordinate skill learning required by the target population.

Any other intellectual skills identified, though not necessarily related to the complex procedures, should also be analyzed to assure that all subordinate skills are identified for learning. A top-down approach is recommended by Gagne and Briggs (1979), in which each subordinate skill identified is, in turn, analyzed to identify its subordinate(s). The portion of the learning map resulting from this phase of analysis is charted as a hierarchy. A designer may have a number of unconnected hierarchies identified at this point in the process, but they will all be integrated into the total learning map later.

Figure 8 shows the hierarchical format for graphically displaying the analysis of intellectual skills. It also shows the superordinate relationship of problemsolving to rule application to concept learning in the model from which this approach is derived (Gagne and Briggs, 1979). Theoretically, these hierarchical relationships are necessary—for example, a rule cannot be learned without the underlying concepts being present in memory—and only intellectual skill is actually incorporated into the new, higher level learning. This structure serves as a guide to designers in conducting hierarchical analysis of intellectual skills.

**Function: Supportive Skill Analysis**

Although no formal model exists to analyze relationships between information learning, attitudes, intellectual skills, and complex procedures, there is some research (Wager, 1976) to help designers identify information and attitudes that support intellectual skill and procedure learning. Figure 9 shows the prescribed facilitative relationship between intellectual skill learning, verbal information learning, and attitude learning in a real technical skills training example. Dotted lines indicate relationships between intellectual and non-intellectual skills, and arrows indicate direction of facilitative flow. Note that non-intellectual skills are presumed to facilitate, and be facilitated by, intellectual skill learning. Facilitative relationships are distinguished from prerequisite relationships in that facilitating skills may not be absolutely necessary for learning, but they are identified in situations where the designer has empirical evidence that the facilitating skills increase the chances that the desired learning outcomes will be achieved for a given target population.
Putting it All Together in a Learning Map

This integration process first requires successive rechecking of the analysis decisions made in each function previously described. For example, after the intellectual skill hierarchies are combined with the complex procedure flows where relationships exist (combination analysis), it often becomes clear upon rechecking that there are learned components which have not been identified in either the hierarchies or the procedure flows. These components and their interrelationships must then be analyzed hierarchically, procedurally, combinationally, or as supportive skills to enhance the tentative learning map.

It is also common to find that what at first appeared to be intellectual skill learning, and was thus subjected to hierarchical analysis, was really in essence a complex procedure with a few intellectual skills supporting its major components.

Thus, in performing instructional analysis at this point, the designer is actually processing procedural, hierarchical, combination, and supportive skill analysis strategies simultaneously in order to identify all the relationships and learned components of the job—that is, combining the pieces that were previously addressed in separate functions.

Clearly, the more experience a designer has in performing instructional analysis, the more skilled (s)he becomes in seeing an overview of the learning map layout for an entire job early on in the analysis process. This overview could be described as prevalent types of learning and the major relationships between them that will form the structure of effective and efficient instruction. Figure 10 shows a portion of an actual learning map constructed at Bell Laboratories for a course on instructional systems design. The section shown illustrates that all four subsets of instructional analysis were necessary to arrive at an accurate graphic representation of the types of learning needed, and their interrelationships, to facilitate job performance for Bell Laboratories course developers.

Conclusion

The instructional analysis approach described in this paper is a critical link between task analysis and the design of instructional objectives for the following reasons:

1. Moving from task analysis directly to the development of objectives is usually inappropriate, because:
   - To achieve adequate performance of many of the tasks and subtasks, instruction may not be required. Unless learning requirements are analyzed, excess instruction may be produced.
   - If prerequisite intellectual skills and/or supportive learning of other types are not derived, the training will be less effective.
2. Performing instructional analysis without first completing task analysis, or trying to do both simultaneously, is ineffective, because:
   - The two processes are performed differently, with different goals and different outputs.
   - Instruction designed without a thorough understanding of the work it is to support will have less chance of positive transfer to job performance. Task analysis provided necessary input to the instructional analysis process.
3. Determining the type(s) of learning (the cornerstone of instructional...
analysis) is an essential activity throughout the instructional design process, because:

- Initially, it helps determine the most appropriate method(s) of instructional analysis.
- It provides the designer with a set of corresponding "conditions of learning" (Gagne 1977) from which to design instructional strategies.
- Both procedural and hierarchical instructional analyses, alone and in combination, should be considered in each instructional design project, because:
  - Using only hierarchies may result in forcing hierarchical relationships that aren't there, or in identification of prerequisite skills at a level too low for the target population.
  - Thinking in terms of complex procedures is more appropriate when the procedural sequence itself is important to learn and when high level decision-making behavior is required rather than expert performance of each step in the procedure.

References


Current Trends In Task Analysis

The Integration of Task Analysis and Instructional Design

Charles M. Reigeluth
Program of Instructional Design
Development and Evaluation
Syracuse University
Syracuse, NY 13210

This paper is about some major trends that are occurring with respect to the development of task analysis methodology. One clear trend is that better methodologies are being developed to analyze cognitive tasks (e.g., Greeno, 1976; Gregg, 1976; Resnick, 1976; Landau, 1983; Scardua, 1983). Among the most promising developments are methodologies for analyzing ways in which knowledge should be structured within a student's head in order to most facilitate given types of performance (e.g., Reigeluth, D. Merrill, & Benders, 1978; Rummelhart & Norman, 1978; Wildman, 1981; Winn, 1978). Much as been written lately about this trend, so it will not be the focus of this paper.

Another prominent trend is that computers are increasingly being used as a powerful tool in helping to perform task analyses. However, many recognize that the important technology here is task analysis methodology itself. The computer is merely a tool for using whatever methodologies we are able to develop. Hence, this trend will also not be the focus of this paper.

Perhaps the single most important trend in task analysis methodology is the integration of task analysis with instructional design; and this trend will be the focus of this paper. But first, it may be helpful to clarify what we mean by task analysis and what we mean by instructional design. Instructional design is the process of prescribing what a specific instructional system should be like. It entails selecting the instructional strategies, including strategies for sequencing the instructional content and strategies for presenting the individual skills and knowledges that make up that content. Therefore, the term "instructional design" is used here in the more common sense of one phase within the entire instructional systems development (ISD) process rather than in the less common, broader sense of the entire ISD process itself. Task analysis is the process of analyzing all the skills and knowledges that should be taught. The purpose of a task analysis is to provide information about the instructional content. Such information may be used for purposes of describing a task, for purposes of designing a test on the instructional content, or for purposes of prescribing instructional strategies (including the selection of enabling skills and knowledges).

Two ways in which task analysis can be integrated with instructional design are: substantive integration and temporal integration. Substantive integration means that the type of task analysis which you conduct is determined by the type of design that you are planning to use, because different types of task analysis provide different kinds of information, and different designs require different kinds of information. Temporal integration refers primarily to having an instructional development procedure in which analysis activities are interspersed among design activities and vice versa, rather than doing all of the analysis and then doing all of the design.

Substantive Integration

With substantive integration, the substance lies in the kind of information that the task analysis produces. The kind of information that is needed as input for instructional design differs depending on the kind of design activity that you are undertaking. For selecting the content that should be taught, you need a type of task analysis that yields the appropriate information for selection to be done well. For deciding how to sequence that content, you need a type of task analysis that yield approriate information for that purpose. For deciding how to synthesize the content (i.e., to teach important interrelationships), you first need to identify the interrelationships that need to be taught, and this requires an entirely different kind of task analysis. (As will be discussed later, this is an aspect of task analysis that has been almost totally ignored, and it is my hope and belief that investigators will begin to address the need for methods of identifying relationships that should be taught. Finally, for prescribing micro strategies (i.e., for deciding what strategies to use to teach a single concept or a single principle, such as the use of generalities, examples, and practice with feedback), a still different kind of task analysis needs to be done to categorize objectives for purposes of prescribing different combinations of micro strategies.

A second aspect of substantive integration is that, with respect to sequencing strategies, it has been found that the selection of a hierarchical sequence (Gagne, 1977) requires one type of task analysis, whereas the selection of a forward chaining sequence (Skinner, 1969) requires a different type of task analysis. And the selection of an elaboration sequence (Reigeluth, 1979a) requires still a different type of task analysis. Therefore, in addition to the fact that each design activity (e.g., selection, sequencing, synthesis) requires some differences in task analysis methodology, the particular instructional strategy that is selected within each of those activities also has important implications for the type of task analysis.

It used to be that most instructional developers would adopt a certain task analysis methodology and use it for all instructional development work they did. One trend with respect to substantive integration is that instructional developers are realizing the importance of having a variety of task analysis methodologies within their "tool kits" and choosing the type of task analysis...
based on the information needs for each design activity (e.g., sequence, synthesis) and of each particular strategy selected during each activity (e.g., hierarchical sequence, elaboration sequence).

Another trend is that, as new instructional strategies are developed, new types of information are often needed to design the instruction, and therefore new types of task analyses are required. As an example, the unique capabilities of micro computers and videodiscs are requiring the development of a new instructional strategies to take advantage of those capabilities. Among the most important of these new strategies are methods for structuring and sequencing the content. There is increasing recognition of the need to design instruction in such a way that a learner can follow his or her interests. For such a "learner controlled" sequence of content (D. Merrill, 1980; Reigeluth, 1979b), some kind of simple-to-complex arrangement of the content is essential; and even for "system controlled" sequences (e.g., sequences controlled either by the teacher or by the computer), simple-to-complex arrangement also have tremendous advantages over the alternatives.

Given that simple-to-complex sequencing appears to be one direction in which design is moving, the question then arises as to what ways task analysis needs to evolve in order to provide the necessary inputs. One problem in answering this question is that there are many different kinds of simple-to-complex sequences because there are many different dimensions of the content on which one can elaborate in the instructional sequence. Therefore, an important design decision is which dimension to elaborate on. In the elaboration theory (Reigeluth, 1979a, Reigeluth & Stein, 1983) we have identified three different dimensions that we think are promising. Each of these dimensions is based on a different kind of generalizable knowledge which is stored in propositional memory. The three different kinds of knowledge are concerned with the how, the why, and the what, all of which are generalizable to new cases. Procedural knowledge provides the how. It is often referred to as operations, procedures, algorithms, rules, skills, and techniques. Theoretical knowledge provides the why. It is often referred to as cause-and-effect relationships, principles, laws, hypotheses, rules, and propositions. Conceptual knowledge provides the what. It is often referred to as concepts, classes, and categories.

When the goals of the instruction call for an emphasis on the acquisition of procedural knowledge, then the simple to complex sequence should elaborate on the procedural content. Information processing task analysis and path analysis (P. Merrill, 1978, 1980; Resnick, 1976; Scandura, 1973) provide very important information for designing that kind of simple-to-complex sequence (see Figure 1). After having conducted an analysis of paths through a rule or procedure, you can design the sequence to teach the shortest path first and then to teach progressively longer paths until all desired paths through the procedure have been taught (see Figure 1 also).

On the other hand, there are many situations in which the kind of knowledge you want the learner to gain is more fundamental, more meaningful in nature, and less along the lines of a rote procedure. This is particularly im-

![Figure 1. A sample of the results of an analysis of procedural knowledge using an information processing task analysis and path analysis.](image-url)
important when there is a lot of variation in the way a task is performed from one situation to another. Such tasks are often called transfer tasks or "soft" skills, and the emphasis in teaching them should be on principles rather than procedures—cause-and-effect relationships rather than sequential steps. For this kind of instruction, the simple-to-complex sequence should be based primarily on starting with the most fundamental, basic principles in a manner similar to Bruner's (1960) "spiral curriculum" and elaborating one level at a time to more complex, narrow and local principles (see Figure 2).

There are also situations in which the emphasis of the instruction is not on providing skills. Rather, the emphasis may be on providing a "general education"—an understanding of the important concepts in a discipline. In this case, the simple-to-complex sequence should be based on conceptual knowledge, initiating the instruction with the most general and inclusive concepts and proceeding through a process of "progressive differentiation" (Ausbéll, 1968) by gradually elaborating, one level at a time, down to the desired level of detail (see Figure 3).

Please do not misunderstand. All three types of generalizable content (procedures, principles, and concepts) are important in practically all courses. It is also true, however, that the goals of a course usually emphasize one of the three types of content, and the elaboration theory merely advocates that that content serve as the basis for the simple-to-complex sequence and that the other types of content be plugged into that sequence wherever most relevant.

It can readily be seen that a simple-to-complex sequence for a course will be very different depending on the dimension along which you choose to elaborate: procedural, theoretical, or conceptual. It should also be apparent that the kind of information (about the content or task) needed for designing a simple-to-complex sequence will be very different depending on which of the three "organizations" is chosen. Therefore, you need to pick a different type of task analysis for each type of organization—you need to analyze a different type of content and content structure. The elaboration theory (Reigeluth, 1979a; Reigeluth & Stein, 1983) utilizes and describes all three kinds of simple-to-complex sequences and also describes the kind of task analysis that is appropriate for each sequence.

**Listing of Principles for Parts of a Physics Course Related to the Behavior of Light**

The most important dimension of complexity for a course on the behavior of light is that the behavior of waves and of particles are respectively less complex than, yet similar in important ways to, the behavior of light. Hence, the simple-to-complex listing of principles is:

- How particles behave
- How waves behave
- How light behaves

The next most important dimension of complexity is that the behavior of light is progressively less complex when it is refracted, reflected, and merely propagated. This progression of complexity can occur within the previous one:

**How Particles Behave**

- **Linear Movement.** They move in a straight line, unless acted upon by something.
- **Reflection.** They bounce off of a surface.
- **Refraction.** They change direction and speed when the inclination of the surface is changed.

(Other behaviors like absorption also come here.)

**How Waves Behave**

- **Rectilinear Movement.** They move in a straight line perpendicular to the wave, unless acted upon by something.
- **Reflection.** They bounce off of a surface.
- **Similar to Refraction.** They change direction and speed when the density of the fluid changes.

(Behaviors like interference, transmission, and absorption appear in turn here.)

**How Light Behaves**

- **Linear Movement.** Light moves in a straight line unless acted upon by something.
- **Reflection.** Light bounces off things.
- **Refraction.** Light bends as it passes from one medium to another.

(Other behaviors like diffraction, interference, transmission, and absorption appear in turn here.)

Just as "How waves behave" and "How light behaves" both elaborate on "How particles behave," and just as reflection and refraction both elaborate on propagation (movement), so the above-listed principles can also be elaborated upon. Space prohibits pursuing elaborations for all of the principles indicated above, so we will pick just one. In pursuing the "Refraction" avenue, the following principles are arranged in a simple-to-complex sequence:

**More Detail on Refraction**

- Effects when light passes from one medium into another
  - image remains the same but apparent position changes
  - rays bend out but remain parallel to each other
- Effects when light passes from one medium into and out of another
  - plane glass
    - image remains the same
    - rays continue in same direction and parallel to each other
  - prism
    - image remains the same
    - apparent location of the image is different
    - rays go off in a different direction but are basically parallel to each other
  - white light is broken into colors (diffraction)
  - concave lens
    - no image or enlarged image
    - rays disperse
convex lens
smallest image before 2FP
inverted image after FP
rays converge at a point, then disperse
Finally, space also prohibits pursuing all of the directions indicated
within the above avenue, one further direction is indicated below. In pursu-
ing this avenue, we elaborate by asking the questions, "What else happens?" and
"Why, which way, and how much does the change occur?"

More Detail on "Into a Medium"
What else happens?

a. A portion of each ray is reflected off of the surface, while the rest is
   refracted into the new medium.
b. The sharper the angle between the ray and the surface, the more of
each ray that is reflected and the less that is refracted.
c. When the angle is equal to or sharper that the critical angle, all of the
   ray is reflected.

Why, which way, and how much do light rays bend at the interface?

d. The higher the optical density, the lower the speed of light.
e. If they pass into a denser medium, the rays bend towards the normal.
f. The greater the difference in optical density between two media, the
   more the light rays bend.
g. Index of refraction \( \frac{n}{c} \) = \( \frac{\sin i}{\sin r} \).
h. Relationship between critical angle and index of refraction: \( \sin i/c = 1/n \).

Why and which way does the apparent size of the object change?

i. When the rays bend, they change their distance from each other.
j. When the rays bend toward the normal, they become farther apart.

Why does the change in the apparent size of the object differ with the angle
of the surface?

k. The more slanted the surface, the more the light rays bend from their
   initial direction.

Principles a through k in Lesson 1 remain of importance, but we can also
add: Why, which way, and how much do rays converge to a point, cross,
and then disperse?

More Detail on "Into and Out of a Convex Lens"

Principles a through k above remain of importance, but we can also add:
Why, which way, and how much do rays converge to a point, cross, and
then disperse?

a. If it passes into a less dense medium, the light rays bend away from the
   normal.
b. On entering glass, rays bend towards the normal by a certain amount,
   and on leaving the glass they bend away from the the normal by the same
   amount.
c. Since the entering and exiting surfaces are not parallel, the normals are
   not parallel, and hence the ray is not returned to its original direction.
d. Since the difference in angle between the two normals increases with
distance from the center of the lens, the amount that rays change their
direction increases with distance from the center of the lens.
e. The more curved the lens, the more sharply the rays converge. The
   image will therefore be larger as long as it is beyond the focal length. Also, the
   focal length will be shorter.
f. Relationship between object size and distance, and image size and
distance:
   \( \frac{s}{d} = \frac{d}{d'} \).
g. Relationship between object distance, image distance, and focal length:
   \( 1/d + 1/d' = 1/F \).

Figure 2. A sample of the results of an analysis of theoretical knowledge
(principles) using the elaboration theory's theoretical analysis procedure.

The following is a brief description of the kind of task analysis prescribed for
each of the three organizations of the elaboration theory. Our purpose in
describing them here is to illustrate the vastly different kinds of information
that a task analysis must yield in order to meet the needs for designing simple-
to-complex sequences of instruction.

Three Kinds of
Task Analysis

Two years ago, the Army's TRADOC commissioned us to try to integrate
state-of-the-art knowledge about how to analyze procedural tasks. What we did
in that project was to take a look at what kinds of information were required to
both select and sequence procedural content. We found that hierarchical
analysis provided one very useful kind of information and that information
processing analysis provided another very useful kind of information.
Therefore, we developed a procedure that integrated appropriate aspects of
both of those methodologies. The resulting product (Regeluth & Merrill,
1981) is called the Extended Task Analysis Procedure (ETAP) because it
extends the existing procedures that were being used in the Army.

Since the Army also expressed some interest in the area of "soft skills," we
also extended this task analysis procedure into the area of soft skills or
transfer tasks, those kinds of tasks which are not easy to procedurize, like
counselling a subordinate. How do you counsel a subordinate? If you tried to
procedurize the task, you would end
up with such an overwhelming maze of
decision steps and branches that it
would be virtually impossible, not to
mention highly cost-ineffective, to use it
or teach it.

Therefore, what ETAP does is to
prescribe a methodology for identifying
the underlying knowledge—the
principles—that can be used to generate the
right procedure for each situation.
Given that underlying knowledge, some-
one can counsel a subordinate about
whatever kind of problem he or she
happens to have. In addition to identifying
the underlying knowledge (principles),
ETAP also provides mechanisms for
identifying decision rules and guidelines
to help the user decide which principles
are appropriate to use at which points in
time.

For analysis of procedural tasks,
ETAP first prescribes that a "process" or
"substep" analysis be conducted, using
The instructional sequence follows a "top-down" approach, teaching each concept at the application (concept-classification) level of learning.

Figure 3. A sample of the results of an analysis of conceptual knowledge using the elaboration theory's conceptual analysis procedure.

an information processing analysis similar to that described by Resnick and Ford (1983). This entails breaking the task down into about six (plus or minus four) steps, including decision steps if appropriate. It then entails deciding if any of those steps are described at the minimum level of entry behavior—in other words if any operations are "atomic" (Scandura, 1983) or "elementary" (Landa, 1983). Note that it is only the operation (or action) that must be at the minimum entry level, not the concepts or facts that are being operated with or upon. For each step whose operation is not at the minimum entry level, ETAP further directs the analyst to continue to break it into about six (plus or minus four) substeps until all steps and substeps have been analyzed down to the minimum entry level of description. The final activity in the "process" analysis entails preparing a unified description of the entire procedure at its entry level of description.

Then ETAP prescribes that a "knowledge" analysis be conducted. This entails identifying any and all concepts and facts that are unmastered learning prerequisites for each step. Again, this analysis continues down to the level of minimum entering knowledge. This is basically a hierarchical analysis, except that "rules" are not one of the kinds of prerequisites that you are looking for now—all prerequisite rules have already been identified as a part of the process analysis. This leaves defined and concrete concepts and discriminations to identify, but discriminations and concrete concepts are seldom analyzed because they are usually part of the students' entering knowledge. Also, we have added facts (Cage's verbal information) as a category or prerequisite knowledge because it is not uncommon for such knowledge (e.g., pi = 3.14) to be prerequisite for being able to perform a rule (e.g., the area of a circle equals pi times the radius squared.)

This completes the analysis of procedural content as prescribed by ETAP. Path analysis is not specifically discussed because it is necessary for one kind of procedural sequence (an elaboration sequence) but not for others (e.g., forward and backward chaining sequences and hierarchical sequences). Hence, ETAP is not specifically tied to elaboration theory. A description of how to continue an ETAP analysis onto into a path analysis and the design of a simple-to-complex sequence is described by Reigeluth and Rodgers (1980).

For the analysis of transfer tasks, ETAP describes a process for helping a subject matter expert (SME) to identify the principles that he or she consciously or more often unconsciously uses to generate the right procedure (performance) for each situation. As mentioned above, all necessary decision rules and guidelines are also identified. Once all the necessary principles are identified, they must be analyzed for prerequisite principles down to the minimum entry level of knowledge. Finally, a knowledge analysis is performed to identify all prerequisite concepts and facts, using the same procedures as for the procedural analysis. This completes the transfer task analysis part of ETAP.

The process for identifying the principles and their prerequisite principles also results in the identification of how fundamental or basic each principle is. Therefore, this process serves the same function as a path analysis does for procedural content, because it identifies which principles are simpler "relatives" of other principles. The analysis of levels of complexity is certainly an essential type of analysis for designing this kind of simple-to-complex sequence, and it is described further in Reigeluth (Note 1) and Sari and Reigeluth (1982).

To summarize with respect to substantive integration of task analysis with the design of sequences, new approaches are needed, especially given such developments in delivery systems as the advent of microcomputers and videodiscs. We need new approaches to sequencing instruction that are able to take advantage of the new capabilities and their requirements for good instructional design. And as these new designs are developed, it is essential that we have task analysis procedures that provide us with the right kind of information for being able to design the instruction properly and efficiently.

Task Analysis for Synthesis

Synthesis is another major area that requires substantive integration of task analysis with instructional design. Synthesis is the process of teaching relationships. The major purposes of synthesis are to make learning more meaningful and to improve retention by creating more connections within one's cognitive structure. You may have had ex-
periences in your own learning where all of a sudden it all seems to fit together; that is a level of understanding that far exceeds the learning of the discrete elements that are taught.

A challenge for instructional design in the future is to figure out good strategies for helping to make it all fit together. In order for that to happen, we've got to be able to identify what relationships need to be taught, and task analysis is the only way to do that. Currently, to my knowledge, there are no analysis procedures adequate to the task. Elaboration theory has identified some kinds of relationships: conceptual relationships, procedural relationships, and theoretical relationships. However, these are not even the tip of the iceberg—they are grossly inadequate for what can and usually should be done in this area. The work of Gordon Pask (1975) and several other cognitive psychologists provide some promise, but as yet such work has not to my knowledge reached the stage of prescribing what different kinds of relationships should be taught to facilitate the achievement of different kinds of goals.

We must know which kinds of relationships are important to teach when, before the methods for analyzing those relationships will be of any use to instructional developers. Much more work needs to be done in this area.

Micro Strategies

Micro strategies was the third major area that I mentioned earlier. In this area, task analysis is done for the purpose of prescribing the best possible combination of micro strategies, including primary components (such as generalities, examples, and practice) and secondary components (such as what characteristics the generality, examples, and practice should have, including visual representations, attention-focusing devices, instance divergence, content attributes presented, and so forth). Those prescriptions are likely to vary depending on whether you are teaching remember-level information or application-level skills; and if they are application-level skills, they are likely to vary depending on the type of content involved: concept, principle, or procedure.

Therefore, what is necessary for good instructional design at the microstrategy level is to classify the various skills and knowledges that are going to be taught. The classification that you use must be one whose categories re-

quote different kinds of instructional strategies. Again, returning to our major theme, there is a need to substantively integrate analysis with design to choose the type of task analysis on the basis of the kinds of information that are needed to design quality instruction.

Temporal Integration

We have discussed how task analysis and instructional design should be integrated substantially—with different approaches for selecting content, for sequencing content, for synthesizing content, and for prescribing micro strategies. The second major way that task analysis and instructional design should be integrated is temporally; that is, instructional design should be initiated before all the task analysis has been completed. Gagne and Briggs (1979) describe a top-down approach to design in which there are four levels of sequencing decisions: curriculum, course, unit, and lesson, in that order. We have recently integrated this notion with our previous analysis and design procedures (Reigeluth & Darwashel, 1982; Reigeluth & Rogers, 1980; Reigeluth & Stein, 1982; Sari & Reigeluth, 1982) to produce an even more comprehensive integration of analysis and design (see Reigeluth, Doughty, Sari, Powell, Frey, & Sweeney, 1982).

These temporally integrated task analysis and design procedures (summarized in Figure 4) start at the curriculum level by (1) identifying the goals and scope of your development effort. This should result in approximately six (plus or minus four) subgoals for each curriculum goal. These will become the course goals as soon as they are (2) grouped into courses. (3) The third step entails grouping the goals into courses.

*1. Identify the goals and scope of the curriculum.
*2. Identify the goals for each course.
*3. Select the scope and organization for each course, and design a simple-to-complex sequence of courses.

For each course:
*4. Identify the goals for each unit.
*5. Select the scope and organization for each unit, and design a simple-to-complex sequence of units.

For each unit:
*6. Identify the goals for each lesson.
*7. Select the scope and organization for each lesson, and design a simple-to-complex sequence of lessons.

For each lesson:
*8. Analyze the organizing content.
*9. Design a simple-to-complex sequence of the organizing content ideas.

For each organizing content idea:
*10. Analyze the supporting content.
*11. Plug each supporting content idea into the simple-to-complex sequence of organizing content.

For each organizing/supporting content idea:
*12. Analyze all unmastered prerequisites.
*13. Design mini-hierarchical sequences by plugging each prerequisite into the lesson sequence just prior to the idea for which it is prerequisite. Next organizing/supporting content idea.

For each idea and fact:
*15. Prescribe micro strategies. Next idea or fact.

Next lesson.
Next unit.
Next course.

*Analysis steps. All others design steps.

Figure 4. Part of an instructional development procedure that illustrates the temporal integration of task analysis and instructional design.
and sequencing those courses. It requires identifying the organization (procedural, theoretical, or conceptual) for the curriculum as a whole (to be used in sequencing the courses) and delimiting the scope of each course. The third step usually results in the allocation of about six goals to each course.

(4) In the fourth step, you analyze each course goal about six more detailed and specific goals, which will become the unit goals after they are allocated to units (next step). You can now (5) group the goals into units and sequence the units. Similar to step 3, step 5 requires identifying the organization for the course as a whole (which will often be the same as the organization for the curriculum) and delimiting the scope and size of each unit. Step 5 usually results in the allocation of about six goals to each unit.

(6) In the sixth step you analyze each unit goal into about six more detailed and specific goals, which will become the lesson goals after they are allocated to lessons (next step). Then you can (7) group the goals into lessons (about six goals per lesson) and sequence the lessons. Similar to steps 3 and 5, step 7 requires identifying the organization for the unit as a whole (which will often be the same as the organization for the course) and delimiting the scope and size of each lesson.

At this point the goal analysis has reached a sufficient level of detail, and the task analysis (or content analysis, depending on organization) begins—but only partly of it. (8) Step 8 directs you to analyze the organizing content for each unit. For example, if you selected organization for a unit, then step 8 entails analyzing and identifying the concepts to be taught in that unit. (9) Then step 9 entails designing a simple-to-complex sequence for the organizing content and modifying the scope of the unit, if necessary.

Once that basic structure or sequence has been identified for each unit (in step 9), you (10) analyze the supporting content. If concepts had been chosen as the dimension for elaboration (the organizing content), then you would need to identify any principles, procedures, and facts that should be learned as well. (11) In step 11, those supporting content ideas are plugged in wherever they are most relevant within the overall simple-to-complex sequence, usually right after the organizing content idea to which each is most highly related.

Now you are ready to (12) analyze the unmastered prerequisites for each piece of organizing and supporting content. A Gagné-type hierarchical analysis such as ETAP's knowledge analysis is most appropriate for this type of task/content analysis. (13) Then in step 13 you plug those learning prerequisites into the overall sequence that has been designed to date, with each prerequisite being included immediately prior to the content for which it is prerequisite.

Finally, you move on to a (14) classification of each of the individual pieces of content that have been selected and sequenced at this point. The purpose of this type of task/content analysis is to prescribe the best combination of micro strategies, such as generalities, examples, practice, feedback, visuals, mnemonics, nonexample, instance characteristics, and so forth. We have found the Component Display Theory's two-dimensional classification based on task level and content type to be most useful for this purpose. (15) Then, of course, you select the appropriate micro strategies to use for teaching each piece of content.

It can be seen in Figure 4 that this development procedure entails frequent alternating between analysis steps and design steps and therefore serves to illustrate the kind of temporal integration of design and analysis that we feel is so important.

Summary

Perhaps the most important trend in task analysis today is the substantive and temporal integration of task analysis with instructional design. Task analysis and instructional design are being integrated substantively by using design activities (such as sequencing and synthesis) as a basis for selecting different types of task analysis, and by using specific strategies (such as a procedurally-based simple-to-complex sequence) as another basis for selecting different task analysis methodologies. The area of synthesis is one that deserves to receive considerable attention in the near future. Also, as new instructional strategies are developed to utilize the capabilities of new delivery systems (such as new strategies for sequencing and synthesis), new task analysis methodologies will be needed to provide the information necessary to design those strategies into the instruction.

Finally, task analysis and instructional design are being integrated temporally by interspersing different task and content analysis methodologies with different kinds of design activities in the instructional development process. It appears that both substantive and temporal integration of analysis and design are very helpful for producing quality courseware.

The unique capabilities of microcomputers and videodiscs require the development of new instructional strategies.

References

Reigeluth, C.M. In search of a better way to organize instruction: The elaboration theory. Journal of Instructional Development, 1979(a), 2(3), 8-16.
Reigeluth, C.M., & Merrill, M.D. Extended Task
The Process of Task Analysis

Integrating Training’s Multiple Methods

Kenneth E. Carlisle, Ed.D.
Nuclear Training Analyst
Palo Verde Nuclear Generating Station
Arizona Public Service Company
Phoenix, AZ 85033

Task Analysis: The term conjures up a magic solution for many training problems. Adherents point out that task analysis can ensure the following advantages (Gibbons, 1980):

- Identification of the full set of training requirements.
- Elimination of unnecessary training.
- Learning and retention efficiency by virtue of logically ordered and organized instruction.
- Improvement of communication within the training development team.
- Reduction in training costs and inadequate job performance.

The advantages of task analysis do seem impressive; however, there are serious problems involved with its implementation. One of these problems concerns definition. There is little consensus about what the term “task analysis” means or how the analysis should be done. Indeed, it seems to mean neither more nor less than the person using it wants it to mean. Lack of consensus on definition is a characteristic of underdeveloped technologies which certainly applies to task analysis (Bernard, 1979). It would seem difficult to consistently achieve the stated advantages of task analysis when little consensus exists concerning what it is and how it should be used.

This paper presents one perspective for defining the process of task analysis and then discusses how various techniques of analysis relate to this perspective. This integration of definition and analysis techniques is essential because training problems are solved by careful analysis, not by undefined and unintegrated task analysis.

Task Analysis:

A Process Definition

What are task analysts trying to do when they examine an area of human performance? The answer to the question cuts to the heart of this definitional problem. Task analysts describe a task in terms of its component subtasks in order to make prediction about the design which will best reduce error in human performance (Davies, 1973a). Ideally, this analysis suggests both how to perform the task correctly and how to instruct a person to perform the task.

The methods applied to describe performance and instruction all use essentially the following process (Patrick & Stammers, 1978; Carlisle, 1982b, 1983):

1. Break the task, content, etc., down into constituent elements.
2. Determine the relationships among these elements.
3. Restructure in accordance with the underlying principle or optimal learning design. (Note 1)

Task analysis may be more properly defined as a process involving these three steps. Different techniques of analysis may be used in different situations for different purposes to describe or design elements of accomplishments and acquisition. These different techniques, however, either follow all or part of the task analysis process, or provide the structure necessary to initiate or complete the task analysis process. For instance, the process of recognizing a potential problem and formulating a deviation statement is one analysis technique which may initiate a task analysis. Analysis of risk may be done to indicate the completion of a task analysis step. These are decision oriented techniques which are integrated with other analysis methods to follow the complete process of task analysis. There is no single technique of task analysis; rather, there are many techniques which contribute to the complete task analysis process.

One final implication of process definition is that the task analysis is not complete until all of the steps are followed. Analysis which stops short of completion often results in bulky documents which remain unused (Carlisle, 1983). However, when all analysis steps are completed in a useable form (for example, a job aid) the results will have immediate impact on training and the workplace.

It is recognized that the process definition broadens the view which some have of task analysis. Some feel that task analysis should only be used to refer to the breaking down of performances into component parts (Kaufman and Mitchell, 1982). Others refer to the restructuring step (Miller, 1962). In this sense the term “task analysis” is unfortunate (Davies, 1973b) because the three process steps previously described can be used to analyze tasks from many points of view, for example, the required content, the knowledge involved, the condition of learning, the required skills, and required algorithmic or heuristic processes (Landa, 1982). Alternatively, the broad process definition offered here might be termed something like “predevelopment analysis” (Gibbons, 1980) or “training analysis” since needs analysis and goal definition are intimately involved throughout the process. For the purposes of this paper, however, the traditional term “tasks analysis” will be retained.

With this perspective in mind, we will examine how task analysis techniques might be most effectively combined and used to break the task down, examine relationships, and restructure for performance and instruction.

Step One:

Breaking the Task Down

Task breakdown begins with the initial definition of the analysis situation. This definition specifies what should be analyzed, when it should be analyzed, and where it can be analyzed. The
following categories may be found in this definition:

- Deviation statement showing what should exist and what currently exists relative to the task.
- General description of the job, duty, or task to be analyzed with references to available job documentation.
- Description of the workers who will perform the task.
- Description of the expected benefits to be realized from the analysis.
- Description of maximum cost to be incurred.
- Description of the preferred format for analysis outcomes.

Once the area of analysis is defined, the analyst breaks the task down with the intention of describing current job and task performance, i.e., how the task is now performed. This breakdown begins with a listing of jobs or tasks—the job/task inventory (McCormick, 1979)—consisting of basic performance objectives for all duties of a job. It is developed from available documentation as well as observations, interviews, questionnaires, or brainstorming sessions with master job performers. From this inventory it is possible to determine which duties require more detailed description by using the following questions to assess the risk of error associated with each duty:

- What is the possibility of an error resulting from this task?
- What are the consequences to safety, cost, public relations, etc., if an error does occur?

The answers to these questions make it possible to select tasks for additional description, since a task with high probability of error and high costs needs additional analysis, whereas low probability and cost mean no additional analysis is necessary. Figure 1 summarizes these decisions.

Additional information might be asked about each task to determine if more detailed description is required. These include performer's characteristics, difficulty of performing or learning the task, importance of the task, performance frequency, number of people involved, time required to perform, where the task is learned, supervision required, and satisfaction gained during performance. This additional information is helpful during the later task analysis steps, but emphasis should be on the probability and consequence of error during task breakdown, since the most critical elements of some jobs (airplane pilots, nuclear power plant operators) are infrequently performed, require little time, and are performed by only a few individuals.

Tasks which require more detailed analysis are then subjected to hierarchical redecision. This hierarchical analysis is used to identify the specific elements underlying the job. The analyst continually iterates the question “How does one do this job?” or “What does this job consist of?” The job is broken down into related duties which are subsequently divided into related elements, and finally into specific motions or procedures. Analysis stops when the possibility and costs of an error for each task or element are acceptable, entry behaviors are identified, or the task is described sufficiently to move to the next step in the process. The technique of hierarchical redecision described by Duncan (1972) and Shepherd (1976) provides a concise format for recording this analysis. It includes the superordinate number, the task component and operation plan, and the reason for stopping analysis.

The Job Analysis Brainstorming Session (JABS) (McDermott, 1982), another method for organizing the process of creating the hierarchy, poses 3x3 cards of master performers' job descriptions on a blank wall. The 3x3 cards show the flow of task performance including alternative descriptions of what is done when something goes wrong with the normal flow of performance.

The initial breakdown describing how the task is currently done may result in very specific task detailing which may be formatted as column charts (Miller, 1962), motion charts, (Seymore, 1968), or flow charts (Merrill, 1980b). Often, however, great detail is not required during this first step because the intent is to provide only enough description to determine likely performance errors and task changes needed to avoid error. Over analysis during this first step often occurs when the actual goal, to organize improved performance, is not stressed. The analyst is not interested during this first step in describing those tasks where errors are likely. The analyst should therefore move to the second task analysis step (relationship determination) as soon as possible.

**Step Two: Looking at Relationships**

The second step in the task analysis process is examining the relationship between the task as it is currently done and how it should, or could be done. The intent is to identify how the task

![Figure 1. Decisions associated with probability and consequence of error.](image-url)
could be changed to remove performance error from the following outcome measures (Jackson and Bullock, 1983):

- Results or effects as indicated by task accomplishment and consumer satisfaction.
- Quantity as indicated by how much of the output is produced.
- Quality as indicated by how well accuracy, completeness, costs, timeliness, and safety are provided in the output.

Four approaches can be used to examine relationships. The existing task might be compared to a predetermined model of correct performance. Different methods of task performance might be compared. Portions of the task that are not affected by performance error might be compared to portions that contain error. Finally, comparisons can be made between individual task components. These approaches are often used in an iterative manner with task breakdown. For instance, a comparison with a predetermined model may indicate the need for a more detailed task breakdown, which may be followed by a comparison with different methods of task performance, which may require additional breakdown.

Behavioral analysis and visionary analysis both suggest models of correct task performance for comparison with the actual task to identify performance error. Behavioral analysis is based upon the systems model or the behavioral model. The input cues encountered, operation performed, information needed, objects used, time involved, decisions required, results and feedback given, and motivations involved are recorded. This information is used to identify performance errors. Three types of error might be found (Duncan, 1972; Reason, 1977; Davies, 1981).

- **Input errors**—those resulting from failure to discriminate a signal or cue leading to an action. The outcome is inappropriate action (also called selection errors or discrimination errors). These are often caused by poor delivery systems that give the wrong information, unorganized information, or poorly timed information.

- **Process errors**—those resulting from neglecting or forgetting a goal, sequence of actions, or piece of essential information (also called response errors, storage errors, or action errors). These are often caused by poor job design, inappropriate tools and equipment, lack of knowledge, or poor motivation.

- **Feedback errors**—those resulting from a faulty verification or checkout of a process. For example, the task is terminated prematurely or continued beyond what is necessary (also called output errors or test errors). These result when consequences and feedback are inconsistent, inappropriate, partially given, incorrect, given for the wrong reason, or given at the wrong time.

The PROBE model (Gilbert, 1982a, 1982b) is probably the most detailed behavioral analysis technique. This model identifies the information, instrumentation, and motivation found in the behavioral environment and in the person's repertoire. The categories of this model are compared to actual task performance using suggested questions to identify where task errors are likely to occur. The PROBE model includes a precise theory about how to restructure the task to eliminate the identified errors.

4. Master performers fill any gaps in the scales with additional incidents. The resulting scale shows how performers with different levels of expertise might execute a given task. This scale can be important when assessing students' entry behaviors.

Problem analysis (Kepner-Tregoe, Inc., 1979) provides a third technique for analyzing relationships. To solve deviations in task performance, this technique utilized the following comparisons:

- What the deviation is versus what it could be but is not.
- Where the deviation is versus where it could be but is not.
- When the deviation occurs versus when it could occur but does not.
- What the extent of the deviation is versus how extensive it could be.

Distinctions between the "is" and "is not" categories are identified and any re-

There is no single technique of task analysis; rather, there are many techniques which contribute to the complete task analysis process.

Visionary analysis (Note 1) is a recently proposed method for creating a correct model of task performance. Visionary analysis avoids focusing upon the problems in the current task by visualizing or mentally rehearsing ideal methods of task completion. Master performers write down a description of what they see, hear, feel, smell, and taste as they imagine doing this ideal task. As much detail as possible is included about who is doing what, when, where, and how. It is then determined if this description is actually what is wanted, and the description is compared with the current task to determine where modifications should occur.

Critical incident analysis is a second way of examining relationships. It is done by comparing different methods of doing the same task. The following steps are involved in this technique (Zemke, 1981):

1. Master performers select and write job incidents and results which are related to specific tasks.
2. Incidents are reviewed for consensus of the master performers, edited as needed, and labeled for ease in rating.
3. Incidents are sorted on a seven or nine point scale for favorable or unfavorable consequences.

Application of these methods to com-
puter assisted instruction where a matrix of relations is used to call up different tasks shows great promise for the future.

In current practice, analysis often use only one method of relationship determination when analyzing an individual task. This is perhaps inappropriate. The use of various techniques combined with additional task breakdown, rather than dependence upon a single method, would give a better understanding of the relationships involved in any given task. Task analysis methodology would certainly be improved if analysts made an effort to combine these different methods before moving to task restructuring.

Step Three: Restructuring in Accordance with the Underlying Principle or Optimal Learning Design

Once the task has been sufficiently described and analyzed for relationships, restructuring for correct accomplishment and learning can be done more easily and exactly. A more detailed description of the improved task should be completed using the results from the previous analysis to develop a job aid (McCormick, 1979), for example, flowchart, checklist, worksheet, decision tree, or decision table. The job aid might be used by employees during task performance, it might be used by designers to change the equipment or tools, or it might be used by management to change the work environment or incentive system. The job aid describes the plan of correct accomplishment, can actually be used in the workplace, and will either eliminate the need for training or will indicate how the training is to be done.

Once learning strategies are developed and sequenced, it is finally possible to develop specific training objectives which not only point to specific behaviors which the learners will be expected to perform, but are also actually required by the task. A master design chart (Davies, 1976), like that found in Figure 3, can be developed as a final component of the task analysis to integrate the tasks and objectives into a coordinated curriculum. This coordination is essential because even though objectives have become prominent in training, there is little doubt that these objectives are frequently not actually derived from complete task analysis or integrated into the overall training design. Objectives can be found at the beginning of lesson plans, but one seldom observes their correspondence to task listing, sources of error, important relationships, learning strategies, hierarchical sequencing, or master design charts. This state of affairs exists primarily because the process of task analysis is so often short-circuited.

Summary

Task analysis is the key to developing the specifications for performance and instructions in instructional systems design. Task analysis, when considered from a process perspective, involves three steps, each of which can be approached with various analysis techniques. These steps and techniques can be summarized as follows.

Step 1. Break the task, content, etc.,

![Learning strategy decision table](image)

Figure 2. Learning strategy decision table.
down into the constituent elements.

Analysis Techniques:
- Task definition
- Job/task inventory
- Risk assessment
- Hierarchical redeployment
- Task tailoring

Step 2. Determine the relationship among these elements.

Analysis Techniques:
- Behavioral analysis
- Visionary analysis
- Critical incident analysis
- Problem analysis
- Network matrix analysis

Step 3. Restructure in accordance with the underlying principle or optimal learning design.

Analysis Techniques:
- Job aid development
- Learning strategy analysis
- Learning hierarchy sequencing
- Training objective development
- Master design chart development

Instruction and training are often derived using analysis techniques which follow this process in a haphazard manner. If task analysis is carefully organized and follows the proper steps, it can be used to solve many training problems—not because of some inherent magic, but because of the detailed, careful, integrated analysis involved.

Reference Note

References


Reigeluth (Cont'd from page 20.)


Winn, W. Content structure and cognition in instructional systems. Paper presented at the Annual Convention of Association for Educational Communications and Technology, Kansas City, April, 1978. (ED 351 315.)

Most of us learned about metaphors in high school. We were told that they were literary devices that allowed an author to lend color and zest to language by speaking of one thing as if it were something else. If that was all there was to the subject, we could legitimately assume that a new book on metaphor would be of interest to English teachers rather than instructional technologists. In fact, the subject of metaphor is very dynamic and of great practical concern to anyone trying to develop communication materials.

Beginning in the thirties, and increasingly throughout the sixties and seventies, semioticians, semanticists, linguists, and philosophers have been arguing vigorously about the role that metaphor plays in meaning (Black, 1962; Fenollosa, 1936; Hawes, 1975; Korzybski, 1933; Richards, 1936; Sacks, 1975). Some philosophers have suggested that our entire language is built up by means of metaphorical concepts. Conventional literary critics have long recognized the metaphorical roots of language, but have always spoken of the underlying layers of metaphor as “dead metaphors” and focused their attention, instead, on the more or less “new” or “unusual” metaphors that add spice to a writer’s style and sometimes cause the reader to experience a flash of new insight. In the last twenty years, however, several lines of research have cast doubt on the idea that any metaphors are really dead. One research program that has produced interesting results has been the study of hypnosis. Successful clinical hypnotists have demonstrated that they can hypnotize even very resistant patients by identifying and using the basic metaphors that underlie the client’s general conversation (Bandler & Grinder, 1975). By skillfully “echoing” the client’s metaphors to gain rapport, the hypnotist can succeed in inducing the patient into a trance. To effect a change in the client’s behavior, the hypnotist may bring the patient’s attention to a new metaphor and suggest that the new metaphor should be used in place of the old.

Lakoff, a linguist at the University of California, Berkeley, and Johnson, a philosopher from Southern Illinois University, have written a wonderful book that boldly summarizes the most comprehensive case that can be made for a metaphorical-based theory of meaning. They argue that metaphor is pervasive in everyday life, not just in language but in thought and action. Our concepts structure what we perceive, how we get around in the world, and how we relate to other people. Our conceptual system thus plays a central role in defining our everyday realities, and our ordinary conceptual system, in terms of which we both think and act, is fundamentally metaphorical in nature.

Consider the metaphor: An argument is a war. (See Figure 1.) Lakoff and Johnson comment:

We don’t just talk about argument in terms of war. We can actually win or lose arguments. We see the person we are arguing with as an opponent. We attack his positions and defend our own. We gain or lose ground. We plan and use strategies. If we find a position indefensible, we can abandon it and take a new line of attack. Many of the things we do in arguing are partially structured by the concept of war.

In Lakoff and Johnson’s view, concepts originate in a multitude of experiences that are related in a family sort of way. Thus, all the concrete experiences that we have of buildings combine to form our concept building. We then use these concepts to metaphorically define still more complex and abstract concepts like, for example, argument. In fact, a complex concept like argument is actually defined or understood in terms of a number of different metaphors. Thus, depending upon our needs, an argument can be like a war, like a journey, like a container, or like a building, or like some combination of these concepts. As a further complication, when we characterize one abstract thing in terms of another, we are not usually relying so much on the concrete images associated with a concept (e.g., guns, headquarters, armies) as we are on the more general processes we associate with a concept (e.g., strategy, battle, victory).

Lakoff and Johnson distinguish between simple metaphors and what they term structural metaphors. Structural metaphors derive from concepts that, in turn, reflect complex experiential gestals. Our experiential gestals derive from the commonplace experiences we all have with our bodies, and with our interactions with our environment and with other people. The common structural dimensions of these experiences include: participants or roles, parts, stages, linear sequence, causation, and purpose. Thus, when a particular concept like war is used, we automatically attempt to structure the metaphorically related concept in terms of the specific
structure our thought in subtle ways. The two concepts connected by a metaphor are not, after all, identical, or they would simply be alternative words for the same thing. Thus, when we choose one concept to describe another concept, we automatically emphasize some aspects of the concept we are describing rather than others. When we use expressions that are based on the metaphor, an argument is a war, we assume antagonists and struggle and a win or lose outcome. When, instead, we use expressions that are based on the metaphor that an argument is a work of art, we do not assume that there will be a struggle and that some participants will lose. Moreover, we assume that the effort can involve creativity and that it may be a valuable and humanizing experience.

We have already mentioned that therapists have become very concerned with patients who employ metaphors that limit their understanding or their attitude about themselves or their environment. As instructional technologists, however, our concern should be with the metaphors we suggest to others whenever we develop written or verbal communication. When we choose an image to begin a slide show—the pairing of visual images is every bit as metaphorical as the pairing of concept words—we are automatically directing the reader’s attention to some aspects of the subject rather than to others. Moreover, since most basic metaphors also imply an attitude, we are, in effect, suggesting the feelings that our reader should summon as he or she begins to learn about a new subject.

One quick example. The best selling sales training program during the seventies was Xerox’s Professional Selling Skills (PSS). Before this course was developed, most sales training was based on retail selling and emphasized tactics that would allow the salesperson to defeat the prospect’s resistance and win the sale (i.e., sales is war). The PSS course was designed for corporate sales representatives whose focus is not so much on a single sale but on the development of an ongoing relationship. PSS introduced the metaphor of sales as a medical consultation. Words and phrases like “professionalism,” “probing for information,” “developing a relationship,” and “prescribing a product mix” all spring easily from this new metaphor. Some instructional technologists would emphasize that PSS was successful because its development was based on a thorough analysis of the basic sequences of behaviors actually used by successful salespeople. It was performance based while many of its earlier competitors were simply based on oration.

But another, equally interesting perspective is that the course was successful because it introduced or promoted a new metaphor that set a better tone and generated a more appropriate attitude toward corporate selling than courses that relied on a metaphor that suggested a win-lose situation. In the words of Lakoff and Johnson:

New metaphors have the power to create a new reality. This can begin to happen when we start to comprehend our experience in terms of a metaphor, and it becomes a deeper reality when we begin to act in terms of it. If a new metaphor enters the conceptual system that we base our actions on, it will alter the conceptual system and the perceptions and actions that the system gives rise to.

Instructional technologists have tended to teach students to analyze and design instructional sequences and assume that they learned writing skills and acquired a sense of style somewhere else. If Lakoff and Johnson are even half right, however, it turns out that ideas and the words and images with which we explain them are so intimately connected that instructional technologists must learn to focus conscious attention on the words and images they use to build their instructional messages.

To raise my own consciousness, I’ve taken to skimming every new instructional program to identify the overall metaphors that seem to lie behind any basic models, rules, or dramatic explanations that are included in the program. Sometimes the language level is too formal and abstract to betray much, but I’m frequently surprised to find obvious metaphors. When I find metaphors I consider each of the structural elements associated with the concepts involved to see how I feel about their relationship to the subject matter, both cognitively and emotionally. What is emphasized and what is hidden by the chosen metaphor(s)? What emotional tone am I being primed to assume? What other metaphors could the author have chosen?

There have been a number of books that have tried to explain the linguistic perspective that I have briefly touched on here, and there are currently several courses, mostly taught by people associated with “neurolinguistic
programming," that seek to teach people how to apply these interesting concepts to practical matters (Dilts et al., 1980). Nothing that I know of, however, comes close to Metaphors We Live By. Forget the poorly chosen title—the book is clearly written, very systematic in its development, and exotically suggestive in its bread-ranging examples. It's a perfect introduction to the role that metaphors play in communicating meaning. It could be used as a supplemental text in an advanced course on design. It would also be appropriate for anyone who is interested in trying to learn to write better. But, one way or the other, this book should be brought to the attention of all serious students of Instructional Communication.—Reviewed by Paul Harmon, Harmon Associates, 3752 Sixteenth Street, San Francisco, CA. 94114.

References


It is perhaps unusual to review a book so long after its publication, but many of the most valuable contributions to our field are published in forms which achieve only limited circulation. This is the case with this volume, so action to broaden its audience is in order.

One of the major recent trends in instructional design has been the development of prescriptive principles based on concept acquisition research. Patterns emerging from that research established the importance of instructional design variables such as structure and sequence of examples, use of negative examples, the role of prompting and fading, and ways of minimizing interference in recall, to name a few. The challenge for instructional designers has been to (in Tiemann and Markle's words), "be practical with theory." We need a system of prescriptive principles and procedures which, when applied a priori to instructional problems of realistic complexity, will produce instruction which is more effective than would have been the case without the use of the prescription.

This book represents one response to this need. It teaches principles and procedures for analyzing an instructional problem and making certain prescriptive decisions for instructional design. These principles and procedures go considerably beyond the conventional objectives-based approach.

The organizing framework of the text is a multidimensional model of instructional goals. Always present is an Emotional dimension. Overlying it are three types of learning: Psychomotor, Simple Cognitive, and Complex Cognitive. Each of these types is further divided into a hierarchical structure. Thus, the Psychomotor learning type ranges from learning of individual responses, through chains, to kinesthetic repertoires. The Simple Cognitive type includes associations, paired associates and serial memory, multiple discriminations, algorithms, sequences, and verbal repertoires. The Complex Cognitive type includes concepts, principles (rule applying) and strategies.

Readers schooled in the ways of Bloom's Taxonomy of the Cognitive Domain may be uneasy with this typology. They may find it tempting to try to pair the levels of learning in the Simple Cognitive and Complex Cognitive with the levels of Bloom's hierarchy. However, it is probably only partly correct to do so. One could speculate that the Simple Cognitive types correspond to the knowledge and comprehension levels in Bloom's terms, and concept learning in the complex Cognitive type corresponds to the application level of Bloom's Taxonomy. However, such a one-to-one correspondence is too facile. The Taxonomy of the Cognitive Domain does not really make the structural discriminations of learning tasks implied by Tiemann & Markle's model. For example, identifying examples of a single concept (a Complex Cognitive task) would probably be identified as no more than an application-level task in Bloom's terms, but following an algorithm (a Simple Cognitive task) would be similarly classified by the Taxonomy. The instruction and testing surrounding the two types of content are obviously different, but an instructional designer using Bloom's Taxonomy might not recognize the difference. It appears that the complexity of Tiemann & Markle's classification is justified by the increased prescriptive precision it offers, when compared to Bloom's Taxonomy.

One might also ask whether the field really needs yet another taxonomy of learning. In this case, the answer is probably "yes." While the matrix Tiemann & Markle construct is unique to this book, the vocabulary incorporated within it is not. Both the terms and their usages correspond fairly well to the language of the research literature. The book helps prepare students to enter the research literature in later studies. This is one model which does not add to our field's Tower of Babble (pun intended).

Within its typology, the book does not address each cell equally. After introducing the model in Chapter One, the authors devote three chapters to paired-associate and multiple discrimination tasks, chain and sequence learning, and concept learning. A final chapter focuses on a technique for analyzing individual concepts for the purpose of instructional design. Psychomotor learning generally is adequately discussed, though it receives less attention than cognitive learning. In the Simple Cognitive hierarchy, most attention is appropriately given to the "simpler" end of the hierarchy. In the Complex Cognitive hierarchy, only concept learning is treated in detail. Learning of principles and strategies is not addressed. The Emotional dimension is acknowledged but not discussed further.

Of greatest interest, perhaps, is the last chapter, on concept analysis. The reader is taught to analyze concepts in terms of their critical attributes (those which must always be present) and variable attributes (those which are present in any of a number of values). Then, a method is presented for systematically constructing sequences of positive and negative examples which will minimize the risks of over- and underspecialization. The general rule is that a range of divergent positive examples should be
presented (i.e., examples which vary as widely as possible on attributes other than those which are part of the concept definition), while negative examples should be as "close in" as possible (ideally, possess all the attributes of the concept, save one). Applying this rule, extremely precise parameters can be specified for constructing sets of positive and negative examples for teaching and testing.

This approach to analysis of content and the design of instruction is a significant innovation. For complex cognitive tasks, the unit of analysis becomes the concept, rather than the skill (or sub-skill), as in conventional (objectives-based) analysis. Emphasis is on the structure of the content to be taught, while the learner's behavior under certain criteria is left implicit. And yet, many of the advantages of conventional skill-based, behavioral analysis have been enhanced. In particular, the prescriptions for teaching and testing frequently are more precise, and the ability to discriminate "need to know" from "nice to know" is in some ways greater.

Although the analysis technique used does not require behavioral objectives, the configuration of stimuli and the discriminations required of the learner are much more fully specified than in a conventional analysis.

Knowledge is composed of networks of concepts, however, not isolated ones. In spite of this, the book virtually stops with the analysis of individual concepts. The organization of concepts into hierarchies is mentioned, and the use of special strategies for teaching related (coordinate) concepts is implied, but neither of those topics is given thorough treatment. This may reflect a conservative decision on the part of the authors, since research into these topics is less well developed than the research into individual concept acquisition. It could be that the authors did not feel justified in going so far beyond their empirical base. Nonetheless, practical design of large blocks of instruction virtually requires use of such techniques. A more extensive treatment might have been justified, even if it required intuitive judgment on the part of the authors.

A second possible area for expanded attention by the book might have been visual concept teaching and learning. As Fleming and Levine have since concluded, concept teaching principles such as promoting on individual attributes apply equally to visual and verbal concepts.

Tiemann & Markle imply this, but virtually all of their examples (even ones with visual content) involve an almost exclusively verbal presentation.

The book is a text, not a review of research. It contains neither references nor bibliography. There is no attempt to relate research evidence for the principles presented. However, given the general caution the authors have obviously exercised in making prescriptive sense out of the research literature, providing references would have been fairly easy. References and a bibliography might have helped provide further guidance to instructors using the text, as well as to advanced students.

These criticisms should not, however, belittle Tiemann & Markle's achievement. The book is effective within the scope established by the authors. For each type of learning, a clear definition is presented, together with a range of positive and negative examples which make the book a model of the techniques being taught. While the general focus is on analysis of content to be taught, the discussion of examples inevitably gets into implications for teaching and testing. This is as it should be; interpretation of the task analysis for design of instruction is often an imprecise, intuitive leap. That is not the case here.

The book is written as a quasi-programmed text. "Quasi" because the frame size is very large (often spanning two or more full pages), and because the inserted questions and feedback seem to be aimed more at reinforcing key discriminations than at facilitating full mastery of complete skills and sub-skills. In addition, branching, pre- and post-testing are not used. Thus, the book is probably not a complete, self-contained instructional system in the sense of a traditional programmed text. However, as a text to be incorporated into an instructional system such as a class, it is well-written and easy to use.

My experience with the book using it both as a text in a university seminar on task analysis and as a major reference for developing practical instructional development procedures in an industrial setting. It has been equally helpful in both cases. In addition to the main text, the book contains nearly one hundred pages of appendices which include extended examples, exercises, and procedures for the various analyses presented. These have proven especially useful in conveying the overall approach to analysis of content.

Comparison of the book to Engleman's errorless learning approach (Engleman & Carnine, 1982) or Merrill's concept teaching system (Merrill & Tennyson, 1977) is inevitable. All three are based on the same body of research, but their approach to interpreting that research for prescriptive use differs. Neither Engleman nor Merrill use a typology of learning as comprehensive as Tiemann & Markle's. They are thus less general in scope. However, the emphasis on sequence of positive and negative examples, and the use of prompting, are similar in all three. Tiemann & Markle and Engleman generally use less ideosyncratic terminology than Merrill & Tennyson. On the other hand, Merrill & Tennyson's step-by-step approach (using the Primary Presentation Forms) is more procedural, while Tiemann & Markle tend to emphasize principles and concepts.

In sum, both teachers and practitioners of instructional design will find Analyzing Instructional Content a practical guidebook for design of instruction using techniques of analyzing and teaching concepts, discriminations, chains and associations, and psychomotor learning. The methods of analysis outlined will be useful supplements—or even replacements—for an objectives-based analysis, it is high time such methods became standard in our field. —Reviewed by Wellesley R. Foshay, Advanced Systems, Inc., Elk Grove Village, Ill.

References

A systems approach was used to assess, remediate, and/or develop compensatory strategies for learning disabled college students. The approach comprises four components: (1) an analysis of the educational task, (2) preparation of criterion measures, (3) preparation of behavioral objectives, and (4) preparation of instructional sequences. A task analysis based on a history syllabus which was done for one student involved the following steps: determining the necessary sequential tasks a nondisabled person would perform to complete the educational task, evaluating the student as to how well the task was learned; designing specific behavioral objectives for the student; and developing an individualized curriculum based on instructional sequences designed to meet the specific needs of the learning disabled student. A copy of the student's posttest is appended.—Microfiche 97 cents, paper copy $2.15 plus postage as document ED 218 839.


Examples of three models for individualizing instruction of library students are presented: Diagnostic Prescriptive Teaching (DPT), Modularized Instruction, and Nonformal Basic Program. The focus of DPT is on general reference tools and subject reference works in science and technology. The model includes four behavioral objectives, a diagnostic test used to determine students' areas of weakness, activities or prescriptions to be assigned to correct the weaknesses identified by diagnosis, and a criterion test to ensure that objectives have been met. A check sheet is included to keep records of each student's progress. Modularized Instruction is used to introduce the student to medical terminology, including the origins of medical vocabulary, often-used prefixes and suffixes, and principal medical abbreviations. A pretest, enabling activities, and a posttest are provided for each of these three objectives. The Nonformal Basic Program teaches conservation of library materials using an eclectic approach. The model comprises four behavioral and three experience-related objectives: selection of needed materials, teacher-made and student-made assignments, sample form for recording student proposal, follow-up activities, and procedures for the monitoring and self-evaluation of the finished product.—Microfiche 97 cents, paper copy $3.90 plus shipping as document ED 221 200.


Developed in response to a need for usable, practical training materials for individualized study while ensuring standardization throughout a large organization, the Qualification Training Package is based on the Instructional Systems Development Model. It can be provided in almost any media mix, and the complete 4-part package provides everything needed in the operational unit, assuming that the needed equipment and tools are already assigned to each unit as part of the operational system. The package includes (1) the Command Job Proficiency Guide, to be used in planning an individual's training to meet both organizational and personal needs, and to document completed training; (2) the Trainer's Guide, which provides instruction on how to use the package for trainers, supervisors, and
The above documents may be ordered from the ERIC Document Reproduction Service (EDRS). PO Box 190, Arlington, VA 22210. Please order by ED number, indicate the format desired (microfiche or paper copy), and include payment for the price listed plus shipping. Inquiries about ERIC may be addressed to the ERIC Clearinghouse on Information Resources, School of Education, Syracuse University, Syracuse, NY 13210.


This report reviews the activities of a project funded by the Bush Foundation to develop resources for improved instruction and professional development of University of North Dakota faculty members. The primary grant activity was the creation of an Office of Instructional Development (ID) and the expansion of existing efforts to enhance the quality of instruction at the university. The new ID office has provided direct assistance to faculty and staff members by providing resource materials on teaching, consultation, and assistance in instructional development projects. In addition to providing a central, formalized source of consultation for faculty and staff regarding teaching, the office has sponsored workshops and seminars, and assisted with ID projects in science education, academic skills, honors, instructional media, and graduate courses related to college teaching. The Bush grants and matching university funds have also made it possible to fund 27 faculty projects involving 33 faculty members, and 39 instructional development contracts in the areas of curriculum, instructional materials, interdisciplinary activities, professional development, department instructional planning, and faculty resources for instructional development. Also covered in this report are advocacy efforts, the organization and governance of faculty development, the results of a formal evaluation covering 18 months of the overall program, the results of questionnaires on the visibility of faculty development work and faculty expectations, and future directions. Microfiche 97 cents, paper copy 55.65 as document ED 219 031.