1999-08-01

Theoretical and Practical Requirements for a System of Pre-Design Analysis

Andrew S. Gibbons
andy_gibbons@byu.edu

Jon S. Nelson

Robert E. Richards

Original Publication Citation

White paper prepared for the Human-Systems simulation Center of the Idaho National Engineering and Environmental Laboratory, Idaho Falls, ID.

White Paper

Theoretical and Practical Requirements for A System of Pre-Design Analysis

State-of-the-Art of Pre-Design Analysis

by

Andrew S. Gibbons
Jon S. Nelson
Utah State University

Robert E. Richards
Idaho National Engineering and Environmental Laboratory

1.0 Introduction

This white paper is the first of two white papers researched and written for the Human-Systems Simulation Center during summer, 1999, examining the nature and theoretical basis for the instructional development process known as *pre-design analysis* (PDA). The second paper (Gibbons & Nelson, 1999) describes a methodology for PDA built upon the theoretical foundation described in this paper.

These two white papers represent a larger program of technological research, presented by the authors at Utah State University. This research is aimed at bridging the worlds of simulation design as practiced by computer scientists and systems engineers and instructional design as practiced by a multitude of corporate, government, and military instructional designers, most of whom lack formal schooling in the techniques of either area.

These white papers judge the issues of PDA from the perspective of the larger instructional design (ID) process within which analysis resides and to which it must contribute important data. Analysis is traditionally separated from design temporally and in terms of the documentation each produces. We have drawn together analysis and design so that the output of analysis is also a design artifact. This analysis methodology anticipates the constructs of design. Analyses conducted using the methods we describe will result in a family of problem structures suitable for use as the structural element for *problem-based* instruction. We feel this is significant because of a growing conviction among instructional design theorists that instruction of all kinds involves the posing of problems in some form to the learner.

The methodology we report here is biased by an instructional theory called *model-centered instruction* (MCI). This theory proposes that what is learned takes the form of complex, highly interrelated, cause-effect or environmental models in the learner's mind; therefore, instruction should be analyzed and structured in terms of constructs that most readily help the learner build those target models. The theory is described in more detail in several sources (Gibbons, 1998; Gibbons & Fairweather, 1998, 2000; Gibbons, Fairweather, Anderson, & Merrill, 1997; Gibbons, Lawless, Anderson, & Duffin, in press).

1.1 The pre-design analysis problem

Jonassen and Hannum (1991) capture the core dilemma of PDA in two statements made at the beginning of a review of task analysis procedures:

> *Task analysis, regardless of how it is defined, is an integral part, probably the most integral part of the instructional development process. All instructional development models to date include some task analysis procedure. . . . Most developers indicate that a poorly executed task analysis will jeopardize the entire development process. (p. 170)*

Yet, Jonassen and Hannum also noted, "task analysis may be the most ambiguous process in the development process. . . . We contend that the ambiguity results from the diversity of procedures and definitions of the process. (p. 170)"

What Jonassen and Hannum describe in their chapter as task analysis is, in reality, a broad family of analysis methods—some directly related to instructional purposes and some distantly related.

We agree with the Jonassen-Hannum assessment of the importance of analysis and the state of the analysis art. Its importance should make analysis one of the best-studied, best-understood, and best-grounded design practices. We find it a cause for real concern that "the most integral part" of the development process can also be rightly characterized as the "most ambiguous".

1.2 Definition of PDA

Discussion of a technological phenomenon rightly begins with describing as many of the problems as possible that the phenomenon is targeted to solve. That is the purpose of this first paper. In this section we reach for a definition of pre-design analysis that describes it in terms that may be useful later in, coming to the most general solutions.

Pre-design analysis is a loosely defined complex of analytic methodologies used by an instructional designer prior to the generation of the details of an instructional design and used ideally as a means for generating those details. Gibbons (1977) and Jonassen and Hannum (1991) provide reviews of a wide variety of analysis techniques, attempting to place them in perspective with the larger design and development process.

Analysis processes involve the construction of hierarchical or networked structures of finely discriminated, mutually inter-linked conceptual abstractions (tasks, propositions, rules, semantic units, schemas, models). The purpose of PDA is to inventory these abstracted units and their interrelationships in a way that fixes them within some type of firm structural framework that gives them constant, measured significance in relation to a deliberated design scheme. Analysis attempts to stabilize the units it identifies so that they can serve as the primitives for designs. One way analysis methods differ from each other is in the significance they attach to these units, which in turn presages the manner in which they will be used.

The interpretation placed on analysis elements and relationships is of great importance. PDA methods have originated from different practical and theoretical perspectives, and the inventors of analyses normally see the analysis as a means to some end. Some PDA methods are purported to capture the names of *tasks*; others purport to capture *knowledge* or an inherent structure of learnable *content*. Many capture primitives from which instructional message will be generated.

It matters whether an analyst sees the result of analysis as a kind of pre-existent *truth* or as a convenient and useful *artifactual invention* of the analyst. In the former case the analyst must defend the truth position and resist change. In the latter case the analyst must see the analysis as an organic and changing data base system in which the value lies in the cunning of the elements and relationships captured. In this latter view, the analysis cannot be considered in terms of "right" or "wrong" but only in terms of relative utility judged by its purposes: element stabilization, element fixation by interrelation, and contribution to design.

Analysis is expected to provide primitive conceptual constructs from which the more familiar instructional constructs—message, strategy, sequence, and interaction—can be derived—generated—through some direct or indirect means. PDA provides the substance of goals and is used to focus instructional events and the structure of instructional environments. Analysis also, therefore, provides the structural basis for the construction of performance benchmarks such as tests and testing scenarios.

1.3 Practical Issues

Though analysis is held in high regard conceptually, the application of analysis in everyday work contexts shows a general *dis*regard (Taylor & Ellis, 1991; Wedman & Tessmer, 1993; Winer & Vasquez-Abad, 1995). Several reasons detailed in sections that follow may account for the disorder in analysis principles noted by Jonassen and Hannum and for the low level of common formal practice. One can begin by cataloging these from a practical point of view. Pre-design analysis for real-world training problems is difficult to perform. Additionally, real people on the job tend to devalue formal analysis when they measure its cost against its perceived benefit.

1.3.1 Practical Issues: Points Of View

The personalities normally involved in some way with analysis decisions include administrators, analysts, and subject-matter experts (SMEs). As an entry point, we can consider the analysis decision from each of these points of view.

1.3.1.1 The Administrator’s Point of View

For administrators the greatest hindrance to PDA is the issue of perceived cost-value imbalance. It is hard to find an administrator who has participated directly in a PDA, but it is easy to find one whose budget has been impacted by it with less-than-satisfactory results. This has two consequences. First, administrators do not understand the details of the process, making it difficult to forecast time and skill requirements for an analysis. When budgets are set, the PDA phase is generally under-funded, creating quality problems in the instructional product. In evaluation, this normally reflects on the cost expended on analysis rather than the inadequacy of the funding, so analysis gets a black eye. Second, administrators who have not participated in a pre-design analysis often are unable to trace the results of the analysis to qualities of the instructional product. This heightens the perception that analysis has little value and leads to the conclusion that it is a waste of time and money.

Two additional factors influence administrator doubts about PDA. Often PDA is performed by mandate from higher administration or from regulatory bodies (Branson & Grow, 1987; *Guidelines for Evaluation of Nuclear Facility Training Programs*, 1994). For an administrator already reluctant to perform PDA, this can create greater resistance to analysis. For the neutral administrator it creates a desire to get the job done as quickly as possible without creating an accompanying understanding of the principles of analysis and analysis links with instructional product qualities.

A similar problem is experienced in the field of Computer Science, where production, at the expense of principle, has often been required in the past. The problem is described by Jim Gray ("ACM Turing Award Presented to Jim Gray of Miscrosoft Research," 1999), inventor of the relational data base:

> *…they [other data base design teams] worked in an ad hoc way; they came to a problem and they solved it, they came to another one and they solved it, too. They could not spend much time on the general properties of these algorithms; they had product to ship. Another group worked at IBM and built the IMS database system that also solved these problems. . . .*
>
> *So there was quite a lot of ferment in this area. People were building systems that actually worked. But there wasn't much discussion about what the underlying theory was or why the systems worked and whether there were better ways of doing things. At IBM Research in San Jose, there was a group of people, including myself, who owe their intellectual heritage to another Turing Award winner, Ted Codd. We were fairly academic in background and more interested in studying systems than actually building them. What I mean by that was we were in*

> *research and were particularly interested in making computer systems that were extremely easy to use. We believed that if a fairly formal theory was the basis of the system, then the system would have much simpler behavior than one with an ad hoc design. I think the success of the relational database has vindicated that approach. (p. 13-14).*

The project Gray describes, because of its attention to theoretical principles, was the occasion for the discovery of the relational unit that is the generative principle of the relational database. The benefit of having discovered this principle has been immense—measurable in hundreds of millions of dollars. However, Gray describes many projects motivated mainly by administrator concern for schedules and budgets overlooked the opportunity—one involving careful attention to a unique unit of both analysis and design.

As a final issue, administrators often find themselves managing staffs who themselves do not comprehend the logic, purpose, and uses of pre-design analysis. This produces additional reluctance to support analysis and an even greater haste to get it finished, regardless of quality.

1.3.1.2 The Analyst's Point of View

Analysts are faced with numerous skill, knowledge, work-style, and attitudinal barriers that build resistance to PDA. Most forms of analysis require some degree of specialized skills, knowledge, and levels of experience. Each method uses a specialized technical terminology and its own logical process. Most analysts are not formally trained instructional designers (IDs) but rather converted subject-matter experts (SMEs). Many are stand-up trainers or technical writers with an interest in technology-based instruction as a career path. PDA is, therefore, most often (numerically speaking) attempted by ill-trained, inexperienced people lacking important skill, experience, and knowledge background. This creates feelings of inadequacy that add to a bias against analysis.

Untrained analysts also tend to be unconfident and hesitant process leaders which disquiets analysis team members. Often this includes a client SME who works for a non-training department. Analysts are thus placed in the position of leading a process with which they feel unsettled themselves in front of a customer.

Many analysts, especially those whose organization is performing analysis under mandate, have seen bad examples of analyses in which the linkage between analysis and instructional design fails in one way or another. Some organizations subvert the process, performing it only to comply. Others perform analysis with good intentions but bad technique, and such analyses can mushroom out of control without creating any product that is useful in later design steps. These factors can cause additional attitudinal problems for already tentative analysts.

The analysis process, which can occur over a period of days or weeks is unusually intensive and can be physically and mentally draining. Analysts who would rather be involved hands-on in materials design—something satisfying and comparatively concrete—rather than analysis—an exercise in abstraction—are prone to the feeling that analysis is unnecessary.

Finally, analysis methods place emphasis on describing correct expert performance but lack special helps to the analyst for identifying potentially risky performance patterns, patterns for detecting and correcting faulty personal performance, or patterns for self-monitoring during performance. This leads some analysts, particularly those who work in risk-intensive fields, to realize that some of the most important results may be omitted even if analysis is performed.

1.3.1.3 The Subject-Matter Expert's Point of View

From the subject-matter expert's point of view PDA can be threatening in terms of self-concept and image among peers. Most experts, no matter how thoroughly trained and experienced, have gaps in their knowledge or are unable to articulate the knowledge they do have. Inevitably the interview process associated with PDA reveals the gaps, and inarticulate SMEs are exposed to a long and unpleasant grilling. For a SME who takes pride in having the right answers, this can be very humbling and embarrassing. However, this is of little comfort to the SME who now appears, in a very public way, visible as less than an expert. This perceived demotion, taken together with the fact that the SME has just recognized knowledge gaps and knowledge that cannot be articulated, creates for the SME the fear that others will no longer consider them expert.

PDA is also perceived as a threat, by the SME, with respect to how others view them as experts. To begin with, SMEs working with an analyst find themselves subjected to a process where they have to relinquish their normal leadership role. The tension that these threats create for the SME can evolve into feelings that the process is too hard and unnecessary.

Finally, the SMEs' lack of knowledge about the instructional development process prevents understanding of how the analysis will be used. This frequently makes it hard for the SME to know how to answer questions that, to the analyst, are quite sensible but to the SME seem to have questionable value. This can be coupled with the fact that most analysts—being neophytes who are learning the content as well as the analysis process itself—often ask what appear to be foolish questions that reveal misunderstandings of things that have already been explained once. The effect for the SME is an even lower confidence in the validity of the analysis process.

1.3.2 Practical Issues: Other

In addition to administrator, analyst, and SME factors that work against PDA, there are other practical factors that hinder many organizations from serious PDA practices.

1.3.2.1 The Process Point of View

Several process issues create barriers for those contemplating PDA. First, analysis is a searching examination of both hard and soft human technologies. It is easy to focus on the more tangible elements of content and miss the truly critical conceptual ones that represent the greatest benefit of analysis. Analysis is an unrelenting exercise of abstraction skills for those who perform it. It is difficult to think abstractly for an extended period of time, even for those used to analysis. And it takes a special discipline to force oneself during analysis to concentrate on the invisible but essential conceptual elements that make the difference between a compliant analysis and an insightful one.

Second, the analysis process causes analysts and SMEs both to think in unfamiliar patterns. Analysis often forces SMEs to cross subtle conceptual category boundaries that are comfortable and useful to the SME but illogical to the analyst. New logic patterns inherent in analytic processes also frequently conflict with SMEs' attempts to match their existing view of the content with one that satisfies the strange and unfamiliar demands of the analysis method.

Third, it is difficult in many cases to see from the beginning how the analysis will produce an outcome clearly related to some tangible instructional product which the analyst and SME do understand. Even for experienced instructional designers it is sometimes unclear how analysis will be used during design, and analysis tends to be highly stylistic and variable from designer to designer. This is at least in part due to a lack of theoretical foundation to anchor analytic practice. There are few principles that restrict or guide changes in PDA methodology. As they are commonly used, PDA procedures tend to be *ad hoc* and highly variable in different hands and produce inconsistent results.

1.3.2.2 The Data Recording and Manipulation Point of View

Analysis practice is hampered by computer tools with interfaces poorly suited to the capture, representation, and manipulation of an avalanche of data that must be organized quickly as it spills forth. The volume of data that accumulates during the process can be enormous, and the uneven speed of the process alternately produces boredom and panic in the analyst. Tools must accommodate not only data recording but the speed at which it proceeds.

Representation is an especially difficult problem, since analysis elements are structured non-traditionally and are often interrelated in multiple, complex patterns. Word processors can be more of a hindrance than a help. Modification of analysis structures without breaking existing relationships is a challenge.

The few computer tools that do exist do not evolve fast enough to keep pace with new developments and new approaches to analysis, and only a handful of relatively traditional methods are represented.

Finally, in the minds of most analysts, analysis documentation is seen as a terminal output document rather than as a living intermediary database that maps forward to specific design elements. Most analysis tools fail to respect this mapping and do not provide for it.

1.4 The Purposes of Pre-Design Analysis

PDA must serve multiple purposes:

1. Primitives for use in design constructions
2. Accountability and requirements tracing
3. As focus/scoping statements for design

Analysis must produce the *primitive transformable artifacts* of a manufacturing process. Analysis in this respect must be viewed as one part of a larger conceptual assembly line where items move along through some transformative process from primitive forms toward articulation and representation. From this point of view, the output of analysis can become an input to the next development stage—design—without further conditioning. This is desirable both from an efficiency standpoint and to eliminate loss of construct validity that inevitably occurs with each transformation. Presently, this principle is ignored, and the output of the most common forms of analysis cannot serve simultaneously as design artifacts. The output of traditional task analysis (TTA) must undergo an intermediate conversion into instructional objectives before it can be related to instructional products. Even then, the tasks from the analysis function as a kind of organizing or title-giving head under which instruction is organized rather than as an element of the instructional structure itself.

PDA should be a means of producing artifacts that take part in a chain of technological intervention processes. Figure 1 shows how instructional artifacts (words, visuals, questions, interactions, experiences) are used to intervene in ongoing human learning processes to influence the path of that learning.

The instructional intervention artifacts used have structure themselves, and the patterns imparted by that structure appear to participate in the learner's own construction of knowledge. The intervention is made with an ideal goal state in mind as an outcome,

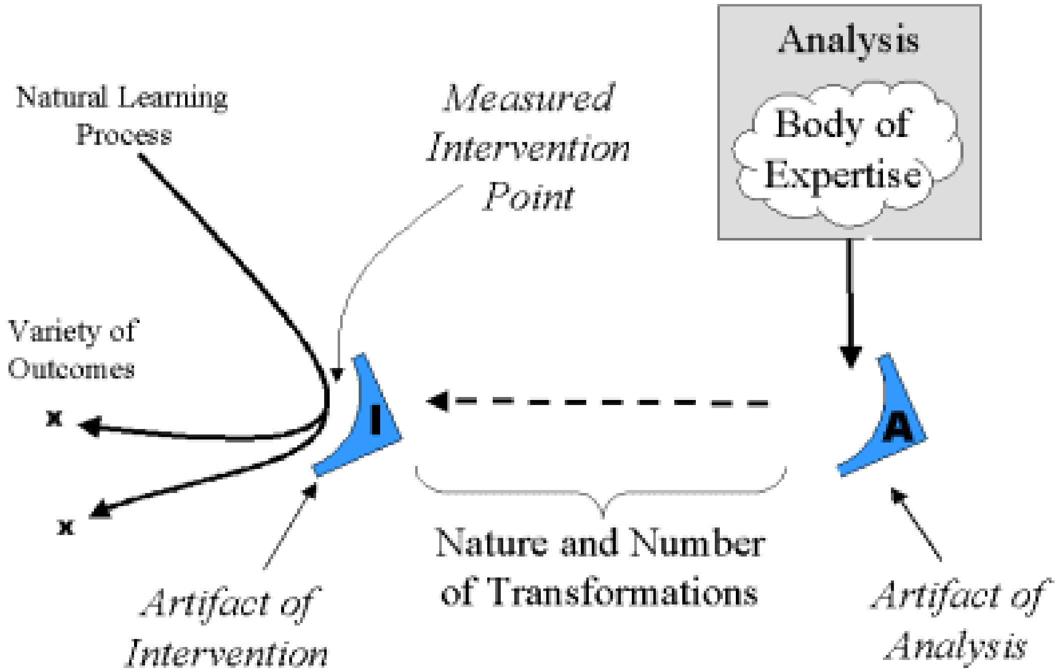

Figure 1. Technological Theory of Analysis

and potentially several intermediate goal states. Some designers interpret the sole purpose of PDA to be the identification of the goal states. Others see the structure of the instructional artifact as the product of other structures expressed together—among the contributing structures being goal structures, information structures, and activity structures. This paper argues that analysis should be a means of identifying all of these.

PDA can also be seen as an accountability tracing process that provides assurance that the subject-matter or domain is completely represented in instruction and yet that only the essential parts of the content are included. As a means of controlling instructional cost, PDA should produce assurances of completeness and minimum sufficiency. Most analysis methodologies, consequently, have a built-in logic that helps to determine when the analysis is complete.

1.5 The Core Mechanism of Analysis

The most powerful PDA processes are engines that generate consistent successive levels of analysis as output from a seed element. The output is normally expressed in terms of structures having the same properties as the seed and arranged in some consistent, information-preserving manner with respect to each other. This repetitive analysis process generates at least a field of seed-like structures and their interconnections with each other.

Therefore, the general case of a PDA methodology consists of: (1) the identification of one or more seed primitive constructs of a given type, (2) the mapping of these input constructs to one or more newly-generated output constructs through a consistent logic of transformation (the output becoming itself a new seed), and (3) iterative repetition of steps 1 and 2 on all seed constructs and outputs until a designated level of analysis is achieved for each one. What differs between varieties of analysis is: (1) the nature and interpretation of the input construct, (2) the nature and interpretation of the transformation performed on the input construct, and (3) the nature and interpretation of the output construct created through transformation.

In a well-conceived PDA: (1) the output construct is of identical form at all levels of the analysis, (2) the output construct is directly usable in design processes in some way, and (3) the transformation set consists of a small number of consistent transformation types. The standard analysis methods that have found wide use adhere to these criteria, but in practice they are violated frequently to the detriment of the output and the confusion of the analyst.

Figure 2 illustrates simply the operational principle of pre-design analysis: input construct-transformation-output construct. Three analysis methodologies—traditional task analysis, cognitive task analysis, and Lesgold's object analysis are compared in Table 1 to show how they fit this common pattern. These three methodologies are each representative of a family of analysis techniques distinguished by the properties of their input and output artifacts and their transformation rules. They are reviewed in detail later in this paper.

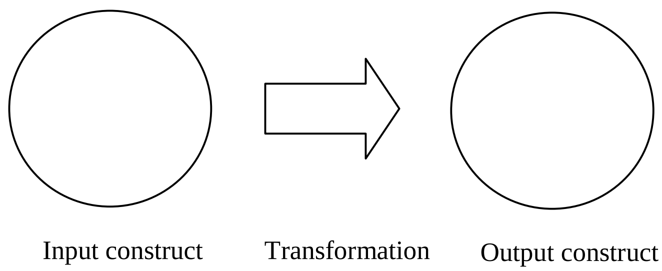

Figure 2. Common operational principle underlying analysis methodologies.

1.5.1 Traditional Task Analysis

Traditional task analysis (TTA), in its many forms, splits tasks in two ways (see Table 1):

1. By fragmenting a task into its major steps

   (take-off → apply power, monitor speed, rotate, depart).

2. By naming the variations of a task

   (take-off → high-performance take-off, short-runway take-off).

Table 1. Comparison of three families of analysis on similarity of operational principle

Op. Construct / Analysis Type	Input Construct	Transformation	Output Construct
Traditional Task Analysis	Overt task	Decompose into subtasks of two kinds: 1. Steps 2. Varieties	Overt task
Cognitive Task Analysis	Overt and covert tasks	1. Break into steps 2. Map to physical and conceptual objects of the action	1. Overt and covert tasks (steps) 2. Physical or conceptual recipient of action
Lesgold's Object Analysis	Expert's conceptual object	Map: -behavior -appearance -inter-object relations	Object world object

Both transformation logics are essentially subtractive: in one the steps of executing a task are subtracted out and captured; in the other the varieties of the task are subtracted out and captured.

The product of a TTA is a hierarchical structure of tasks. TTA begins with a seed that represents the most inclusive (highest-level) task and generally identifies at least overt or observable tasks performed in the real world during task execution. Consistent application of the steps-varieties transformation produces ever-smaller task artifacts as outputs that also tend in real applications to be overt tasks. This shrinking output construct is assumed to identify acceptable target real-world performances that can be used as instructional goals (after conversion into instructional objectives). As more detailed levels are reached, tasks are presumed to be of a more practically teachable and testable size. Gagné (1968, 1985) popularized the design principle that lower-level tasks, once mastered, enable practice at more integrated levels represented by tasks higher-up in a hierarchy of decomposed tasks.

1.5.2 Cognitive Task Analysis

Cognitive task analysis (CTA), like TTA, also breaks overt tasks into smaller tasks beginning with a higher-level seed. But a major distinction is that CTA especially targets covert, or invisible, tasks as well. These tasks are presumed to be especially important to expert performance, and CTA is intended as an analysis of expert performance. Covert tasks include thought processes, judgments, and decisions of an expert. Unlike TTA which concentrates on the outward performance, CTA looks at both the outward and inward performance with emphasis on the inward.

A second major distinction of CTA is that it maps, in some fashion, some parts of the physical, conceptual, and functional systems that are the objects of expert action (see item #2 of Transformation column in Table 1) and connects these objects with tasks in some way. CTA is used to identify physical, conceptual, and functional properties of the system on which the expert performance impinges. In most CTA analysis methods, however, this is done informally or as a secondary part of the task specification. Although, CTA emphasizes covert tasks and adds object-to-task linkages, the task-to-task transformation (see Figure 1) that characterizes TTA is true for CTA also.

The work of R. B. Miller (1963) is a pre-cursor to cognitive task analysis. Miller, an information-processing psychologist attempted to capture and typify specific *information processing patterns* through analysis. Most current CTA methodologies focus on information processing that is particular to expertise in a specific subject-matter and do not try to identify general patterns of cognition.

1.5.3 Lesgold's Object-Based Situational Approach to Task-Analysis

Lesgold's object-centered analysis is a third family of approaches to PDA that contrasts sharply with TTA and CTA, which are both task-centered. No examples of its direct application are known to us, but its influence on Lesgold's recent work is clear (Lesgold, 1993).

Based upon the principles of object-oriented design, Lesgold suggests analysis where domain knowledge "is represented by a collection of…objects". An object is "an entity that has states, behaviors, and identity". Contained with the object, these properties and behaviors describe how objects interact with each other, sending and receiving force and information, and changing within themselves. The input for Lesgold's analysis is not the real-world object but the objects in the performance perceived by an expert performer; the output is an object described in an object world. The transformation in this analysis is a mapping from the expert's conceptual world to objects in an object world that mimic the expert understanding of a system.

The object analysis process is not a fragmentation of a system into components in the sense that TTA and CTA fragment tasks, though hierarchies of objects may result because in the expert's world objects tend to be composites of component objects. The object world constructed by this analysis corresponds with the expert's world only in terms and units the expert finds useful, and as Lesgold describes in recent work, "our schematic diagrams are designed to reflect how expert technicians think about the system. The grouping of components in the diagrams matches expert groupings" (1999).

These output constructs are themselves directly convertible artifacts for design, not in the sense of creating performance goals but in the sense of describing an interactive system that can be used to provide interactive student experience. Lesgold describes an instructional setting where these objects can be placed in an environment with like objects that interact with each other and with a coaching expert that monitors the objects as they interact and generates instructional messaging. Ultimately, for Lesgold, the objects encapsulate expert “knowledge” to be trained. They secondarily encapsulate overt and covert expert skills but primarily the conceptual properties of the objects experts act upon.

1.5.4 Comparison Summary

The preceding examples demonstrate how the operational principle of PDA shown in Figure 2 facilitates understanding of apparently diverse analysis approaches. For the remainder of this paper this underlying explanation of analysis technology will be assumed. This principle allows us to look closer at any process of pre-design analysis and ask questions that otherwise might be overlooked. These are the questions that need better answers in order for analysis technology to advance.

According to this principle, differences in analysis process can be categorized as variations in input construct, the nature of transformations, and nature of output construct, or some combination of these. Given this principle, some immediately relevant questions come to mind: (1) How does the choice of PDA input construct influence the type(s) of transformations possible? (2) How do the PDA transformations govern the type of output construct produced? (3) How do variations in any of the three components (input construct, transformation, output construct) influence the desired “assembly line” process of design and increase or decrease the possibility that the output construct can become a ready-made input to design without further conditioning? (4) How do variations in any of the three components influence the types and quality of instruction ultimately produced? (5) What new analysis methodologies are suggested by this operational principle to open up the range of possible analysis-design linkages?

1.5.4.1 Taxonomic Systems

One very popular variant of this analysis principle is to assume a limited set of output constructs. This leads to the development of taxonomic systems that itemize possible or allowable output constructs. The arguments for making this taxonomic assumption vary from theoretic to practical. Theorists argue for particular sets of analytic output constructs on the grounds that they correspond with human aptitudes (Guilford & Hoepfner, 1971; Horn, 1989; Kyllonan & Shute, 1989), define basic learning types derived from classical research categories (Gagné, 1985; Melton, 1964), or represent constructs related to human performance (Anderson & Lebiere, 1998) or human information processing (Miller, 1971). Gagné's taxonomy (compare editions 1-4 of *Conditions of Learning*) shows maturation over time as the dominant learning paradigm shifts from behaviorist to cognitive (Gagné, 1965).

Starting with a practical concern, Bloom and his associates (Bloom, 1956) constructed a taxonomy whose original intent was the categorization of test items. Designers found that Bloom's categorization scheme also supplied a set of analytic output constructs, which is the use most commonly made of the categories today. Merrill (Merrill & Twitchel, 1993) has organized several versions of analysis taxonomies that attempt to directly link with instructional treatment variables, the long-term goal being reduction of development time and cost, improved consistency of the instructional product, and eventually automated development. Regian (Regian, 1999) has devised a taxonomic system to speed analysis that leads directly to the design of intelligent tutoring systems (ITS). His work shows the crossing of the taxonomic mechanism into the area of ITS that in the past has been known for its hand-crafted, one-of-a-kind analyses.

1.6 Historical development of Pre-Design Analysis

The present practice of PDA can best be understood in terms of the history of PDA methodologies. Earlier, more traditional PDA methodologies were not derived from theoretic bases: they were expediencies to meet rapidly growing training needs brought about by the increasing complexity of human work and work systems. Most early analyses were created out of practical necessity. This realization becomes significant as a researcher tries to understand the underlying principle of an existing analysis methodology or extend it beyond its current usage.

1.6.1 Traditional Task Analysis

The beginnings of formal task analysis are commonly acknowledged to be in the work of Frederick Taylor and his attempts to devise principles of scientific management. Taylor used the analysis of work tasks to quantify them and determine their contribution to efficiency. Over time it became apparent that the task analysis had more uses, particularly in job definition and description and as the basis for training individuals in their jobs. From the beginning, however, the task and its analysis were of practical, not theoretical interest and were used mainly as a basis for measurement.

The need for better and more efficient training during World Wars I and II and subsequent hot and cold wars through the fifties and sixties combined with the enormous and growing complexity of rapidly proliferating electronic systems of all kinds to boost the importance of practically-oriented task analysis. Without stopping to build a theoretical basis for analysis, designers increased the number of new methodologies.

Heavy demands for analysis product and improved methodology resulted in ample discussion of the “how” of analysis but did little to shed light on the “why”. Gibbons (1977) made an extensive review of analysis methodologies prior to 1975, and Jonassen and Hannum (1991) have reviewed methodologies of more recent date. During this period of great expansion, academic centers for the study and propagation of analysis were established—Ohio State University and UCLA being two prominent centers—task exchanges were set up, and many organizations of the military and government adopted task analysis as a standard with multiple uses.

Out of the increasing complexity of military and industrial systems there came a need for new and more sophisticated ways of viewing, planning, and representing combined human-system functions, leading to the emergence of general systems theory. The new “system” thinking significantly influenced all fields in some way but was especially influential among subsequent generations of instructional designers. It led to multiple views of “systematic” approaches to instructional design (Banathy, 1987; Branson et al., 1976).

As part of this general movement, many organizations, particularly the military, considered the analysis of tasks to be important for training purposes aside from its administrative uses and began to formalize and institutionalize it through process specification and regulation. Perhaps in an attempt to help build a theoretic base, while at the same time improving process efficiency, controllability, and consistency, part of the expansion of analysis methodologies through this period involved attempts to devise the performance or learning taxonomies that have already been described.

The rise of radical behaviorism in the period of the sixties also encouraged analytic techniques. B.F. Skinner (1953) used the *operant* as a unit of behavior analysis. Analysis in this paradigm involved decomposition of a performance into individual operant (S-R) units and the identification of "chains" of operants that constituted more complex performance. Many writings appeared at this time for guiding instructional designers through this new analysis process, which was closely linked with the programmed instruction movement (Glaser, 1965; Lange, 1967). Cook (1997) describes problems that arose as analyses were conducted by program designers with different judgment and understanding of the operant principle. Though some writers have identified behaviorism as the main cause of the rise in analytic methods (Schraagen et al., 1997), given the history recounted so far, behaviorism can be seen as just one of several influences that reinforced trends already underway. For instance, analysis also appealed to those attracted to the broader trend of structural and "systems" thinking that was rising during this period.

Use of task analytic methods for pre-design analysis accelerated during the '50s and '60s, influenced by large system design projects like the DEW Line (Distant Early Warning radar) that involved complex electronic designs, enormous variety in trained skills and positions, and complex, high-risk training to high performance standards. Such systems required large numbers of trained personnel to run, maintain, and administer them—literally hundreds of jobs—and, the training for many of those jobs overlapped or was repetitive. A trend that developed was the derivation of training analyses from operational systems analyses that were required of government contractors as part of their system documentation. Though more recent insights into human performance and cognitive dynamics seem to have discredited this approach to analysis, good arguments can be made for retaining system components and functionalities in a complete analysis. This issue is taken up again in a second working paper.

Increased use of formal pre-design analysis stimulated a proliferation of analysis guidelines and standards, particularly within the military. The culminating point and most visible artifact of this standardization is the Interservice Procedures for Instructional Systems Development (IPISD) (Branson et al., 1976), an instructional design and development model intended for use by all military services as a procedural guide for untrained instructional developers. IPISD described a simplified analysis and design approach intended for use by a large audience of lay designers.

Pre-design analysis became general throughout the military as a process standard because: (1) IPISD and numerous other analysis standards like it were enforced, (2) IPISD and other clearly-expressed process standards made analysis activities accessible to novices, and (3) enforcement took place over an extended period of time. The acceptance of task analysis methods became general also within government agencies responsible for regulating high-risk industries (e.g., aviation, nuclear, aerospace). Through this period, the generally atheoretical nature of analysis did not change.

1.6.2 Cognitive Task Analysis

Over time, the complexity of electronic systems was replaced by even more challenging computerized systems. It became difficult for training to deal with the complexity of the systems that students were asked to operate and maintain. For instance, in aviation, the increasing use of modular line-replaceable units (LRU) to facilitate aircraft maintenance led to the construction of complex test benches where LRUs reported broken and replaced could be tested for faults diagnosed. The LRUs were complex enough to defy complete understanding by their maintainers, but the test benches represented an even greater level of system and functional complexity.

Instructional psychologists at this time (early ‘80s) were becoming aware of a shift that had long been taking place in the nature of the tasks being trained—from overt and observable to covert and invisible. This shift seemed to call for the development of more detailed and complete analysis technologies. Training doctrine to that time had recognized only the need for an inventory of job tasks, but it became apparent that a form of analysis that captured expert reasoning and decision processes was also required.

This realization prepared the way for a methodology called cognitive task analysis (CTA). As the name seems to suggest the new analysis capitalized on and extended the existing mechanism of task analysis. CTA did this by grounding itself in assumptions about human expert performance and in doing so connected itself with an extensive body of cognitive theory and research. CTA became the subject of intensive study among an intellectually active group of researchers studying artificial intelligence and intelligent tutoring systems (Wenger, 1987).

The development of cognitive task analysis methods profited greatly from research into intelligent tutoring systems. The grail of these researchers was the automation of the process of instructional design by placing design at the moment of instruction. This generative rather than pre-constructive approach seemed to solve the problems of complexity and volatility of subject matter, at the same time reducing development and maintenance costs. Work begun by Carbonell (1970) on machine-representable content matured into a well-funded and broad stream of research by a community of psychologists and computer scientists through the late 60s, 70s and early

80s. This work reviewed by Wenger (1987) continues to be of interest today, funded mainly by military and government agencies (C. Youngblut, 1994), and summaries of the effectiveness and cost of this kind of instruction have been positive (Fletcher, 1999).

A range of instructional forms was devised and tested, each heavily invested in an approach to PDA that differed from traditional task analysis. Since the motives of this research also included investigation of cognitive processes of learning and their instructional theory implications, analysis at this time took on a theoretical implication that could not be avoided, and questions regarding *knowledge* and the possibility of its external representation became relevant. Intelligent tutoring systems tended to center on what researchers thought was the representation of *expert knowledge* and corresponding *knowledge states* within the learner. This connected analysis with theory and re-opened interest in epistemological issues that continues today. Researchers intensified their study of the implications for analysis of the real-time generation of instructional messages and interactions.

Cognitive task analysis has emerged from over two decades of research as an umbrella methodology that can best be characterized as a constellation of individual techniques formalized in slightly different ways for use by different audiences of instructional designers. Some CTA methods can be applied only by trained and experienced analysts. Others, like the PARI method (Hall, Gott, & Pokorny, 1995; Youngblut, 1994), have been devised for a wider and less experienced audience. In general, however, CTA remains today the province of trained and experienced analysts because despite simplified methods like PARI, the average designer has neither the time nor the motive to deal with the complexities of the analysis at such a level of detail.

CTA has remained mainly the tool of the ITS designer. Publications on CTA have reached a wide general audience (Merrill, 1987; Psotka, Massey, & Mutter, 1988). These have been, in large part, the appeals of academics who hope to bring the methodology to a wider audience of users, who in most cases do not see a connection between the more extensive analysis and an improvement in their designs. Widespread formal application of CTA methods among average designers is not evident.

CTA has a grounding in theory that traditional task analysis does not have. It may be more correct to say that each variation of CTA has its own theoretical assumptions and implications. Sufficient data has accumulated (Fletcher, 1999) to establish that intelligent tutors based on the general principle of CTA are effective instructionally and cost-competitive as well. This would appear also to validate CTA as a methodology through its contribution to the instruction.

The success of the methodology may mask important theoretical questions that have not yet been fully explored, however. Since CTA methods tend to be based on cognitive science and learning theory roots, a reasonable—we believe harmful—conclusion from the positive results might be that they capture something that could be termed “knowledge”. This is the epistemic question related to analysis. The next section addresses this issue by examining four analysis methodologies from ITS research.

1.7 Review of Selected Analysis Methodologies

For each of four methodologies reviewed in this section, we are interested in answering:

- What is the methodology’s theoretical base?
- What assumptions are made by this methodology about knowledge?

- How does the technological principle (input-transform-output) of analysis apply?
- Why is this methodology important?
- Does the practice of this methodology measure up to its stated principles?

We have chosen methods that, with one exception (Lesgold), have been tested in documented applications and demonstrated to produce acceptably effective and efficient instruction. All of the examples have been chosen from the tradition of intelligent tutoring system research.

1.7.1 Schank

Schank (Edelson, 1998) uses subject-matter expert-supplied *stories* as an analytic construct that crosses the analysis-design barrier. In *Tell Me A Story* (Schank, 1998) and *Inside Case-Based Explanation* (Schank, Kass, & Riesbeck, 1994) Roger Schank describes mental processing in terms of schemata—structural frameworks—that are used for the acquisition, organization, and application of personal knowledge. Stories as an analytic construct bridge an analysis-design gap for Schank and provide as an output an artifact that can be directly applied to design.

Analysis by Schank proceeds through subject-matter expert interviews in which the expert is asked to contribute stories of problem solving during expert performance. A large volume of such stories is acquired, and content appears to be largely at the discretion of the SME. Acquired stories are then subjected to extensive indexing—akin to the subject indexing of library resources but painstakingly more detailed—using a master set of indexing descriptors (Schank & Fano, 1992). This master set represents a comprehensive semantic description of human-object and human-human interactions.

The output of analysis is a set of indexed stories whose indices can be used to draw forth an instructionally useful and context-appropriate story at any time on learner or system demand. The goal is that the story be related to a question or need of the learner, who is engaged in a problem solving activity. Because the unit of the analysis—the story—makes direct contribution to construction of instructional messages, the output of analysis is directly the input to design.

Schank has selected an analysis unit with practical utility, but the unit is also an element of his theory of learning by problem solving (Schank et al., 1994). The stories themselves represent a form of capture for expert behavior associated with each problem situation, and during story gathering, multiple viewpoints are elicited from multiple experts. But there is no process for formal definition of the environment, the systems, or the expert behavior associated with each problem. So at some point there must be additional analysis that is not described in Schank's literature to accomplish this. The issue of completeness also faces the analyst who must somehow decide when an adequate number of stories has been gathered and indexed for a given family of problems. The story analysis therefore produces a good range of expert viewpoints on a problem, but the methodology provides no means of determining when an analysis is reasonably complete other than the consensus of expert reviewers.

One of the most important and interesting elements of Schank's analysis is the taxonomy of indices used to cross-reference stories to learning questions (Schank & Fano, 1992). Schank and his associates have expended immense effort creating and testing this list of descriptors devised to index a story's main "points". Working manually, indexers can cross-reference stories containing any subject matter or content in terms of the dramatic structure and details of the story. Indices characterize the story in terms of goal-directed actions by intentive agents having particular values and beliefs and summarizes their success in attaining the goal along with an explanation of the reason for success or failure. An index allows stories with different content to be matched in terms of these internal structural properties. A typical string to index a story about a receiver who is able to catch the football because a defender trips and falls might be: *"the execution of plans may be affected favorably by conditions on the plan that are fortuitously met"*. The same index would be appropriate for a story about a fire-fighting team that is able to control a forest fire due to shifting wind conditions.

Schank's work provides an interesting departure from traditional approaches for grounding pre-design analysis. Rather than capturing what might be termed "knowledge", Schank's method captures episodes in which the residue of expert knowledge-using processes are encapsulated. The student must extract from multiple stories the raw materials for constructing personal knowledge.

1.7.2 Lesgold

Lesgold's research has tended to emphasize—as most ITS research has—the construction of expert performance models. His description of *object analysis* is a departure that explores an under-emphasized alternative for analysis. Objects are the output construct of Lesgold's object analysis (Lesgold, 1993). The input is a description of a real performance setting filled with objects to be identified.

Lesgold's use of objects as the analytic unit may be stimulated by the rising popularity of object-oriented programming and the availability of more or less user-friendly object tools. He may have considered this type of analysis a means of matching the nature of the instructional construct with the programming construct. Alignment of conceptual and tool constructs is presented as a major issue in instructional design by Gibbons et al. (Gibbons et al., in press).

The use of objects may also be an exploration of methods for implementing ideas in Lesgold's more recent work (Lesgold, 1999). In it he restates earlier themes of Kieras (Kieras, 1988) that emphasize careful choice of level of depth and resolution in the representation given to the student of system models. Particularly, these ideas urge the simplification of the detail in those models to match the level of expert performance applied to them, rather than giving the learner models with excesses of irrelevant and useless (for performance decisions) information. Lesgold's object analysis, by dividing the experiential world into independent parts, makes it possible to analyze and represent in detail those parts of the world that the expert can influence through decision and action, while presenting in less detail those that are unrelated to user decisions.

Lesgold's analysis is based on the operational principle of capturing "things" as opposed to capturing "knowledge". Lesgold also discusses capturing multiple versions of objects at different levels of detail and resolution to support growth of learner expertise over successive stages of complexity. In this respect, Lesgold's ideas echo research by White and Fredricksen (1990) on the evolution of progressive externalized teaching models as a means of fostering student creation of a parallel series of successively complex internal models of a system.

Lesgold discusses the possibility of using lexical techniques in connection with objects. This is similar to Schank's use of sentence-like indexical strings as an analysis tool. Terminologies are shared by experts as an entry point into both theoretical and practical aspects of the expert's world. Using the expert's nouns and verbs (terminologies) as an entry point to analysis seems to us especially appropriate. It suggests among other things a form of analysis like that used in the 60's to inventory electronic maintenance tasks in which the tasks performed varied little across systems, compared with much greater system-to-system differences.

Task analysis for such systems was in some cases referred to as *task listing*, and the output was a table that crossed tasks with components to which the task was to be applied. Table 2 gives an example using fictitious content to illustrate:

Table 2. Illustration of the lexical approach to analysis of electronic maintenance tasks.

	Test	**Replace**	**Repair**
Azimuth module	X	X	
Receiver module	X	X	
Display module	X	X	X

An "X" within a cell designates a task for training; a blank cell removes the task from training. Analyses of this type were often performed largely as a matrix construction exercise in which the real determinant of the analysis was in the lexical units (names of components and action verbs) selected to head rows and columns of the matrix. Lesgold's discussion of lexical techniques during object analysis is reminiscent of this technique.

Lesgold, however, proposes that using objects as analysis constructs entails also collection and cataloging of object information, including properties and behaviors as well as resources for expression of objects. This corresponds with the work of several researchers into the uses of self-governing objects in instruction, including Repenning (Roschelle et al., 1999), Resnick (Resnick, 1997), Papert (Papert, 1980) and others.

Merrill (Merrill, 1999) has described a methodology of "knowledge objects" that has been adopted and adapted by a wide audience of tool makers and large-scale tool users (NETg, Macromedia, Oracle). Drake (Drake, 1997), whose work is described in more detail later, used objects as the basis construct for an authoring tool for instructional simulations. Drake's tool allows non-programmer designers to generate system models from objects and instructional message and interaction from raw object properties.

1.7.3 Regian

In pursuit of the automated design of intelligent tutoring systems, Regian and his associates at the Air Force Research Laboratory have evolved classes of learning that prescribe instructional activities that automatically pre-condition the PDA (Regian, 1999). Regian's learning categories are offered as the distillate of decades of cognitive and information processing research. This theoretical derivation is similar to the category systems proposed recently by Gagné (1985), Merrill (1999), Bloom (1956), and earlier by Guilford and Hoepfner (1971), Miller (1971), and others. Regian's system, is therefore, one representative among several taxonomy-based systems, and, in this case, the expert's "knowledge" is presumed to be represented in the analysis product. Regian's method is of special interest because instructional design principles associated with each category have been described in enough detail to be embodied within an intelligent tutor framework, and several instances of tutors have been field tested. The resulting instruction appears to be effective, and additional benefits of reduced instructional time and reduced design and development cost also seem to be gained.

The input to Regian's methodology is the unorganized (for instructional purposes) personal knowledge of the subject-matter expert or expertise resource (manual, technical documentation, etc.). The output is a list of categorized instructional goals and, eventually, the content to support their teaching. One of the purposes of the automated design system is to guide the extraction of content relevant to specific instructional goal types. In a fully-articulated design system of this type, in principle the designer concentrates solely on the analysis of the primitives inherent in the subject matter and never need see or deal with compiled message parcels, displays, resources, or instructional sequence plans. How well that ideal is attained in automated design systems varies from system to system. Regian's system avoids the problem of fragmentation that is a common objection to taxonomies by defining categories that are cumulative and that represent not only accumulation of knowledge but stages of performance through integrated and automatized levels.

A system like Regian's in which instructional message and interaction are generated from input primitives makes the relationship between analysis and design explicit and traceable. Specifically, it shows how the nature of the demanded input and the nature of the transformation to which it is subjected determine the range of output possible. More importantly, it shows how an analysis is tempered by its intended use. Instruction that constructs instructional message from sets of primitives must pay special attention to the nature of the primitives and make arrangement for their generation during analysis. Instructional designers traditionally think in terms of the use of analysis output to determine key factors of the design. Systems like Regian's show how design standards can also exert an influence backwards to determine analysis categories.

1.7.4 Anderson

Anderson (1993) describes a cognitive theory, ACT*, that translates directly into instructional theory and has served as the basis for the development of a family of intelligent tutoring systems. Anderson's theory is based on the hypothetical construct of the *production rule*: "an encoding of knowledge that is optimized for use" (p. 31). Production rules (referred to in shorthand as "productions") consist of a condition (IF) part and an action (THEN) part. Anderson's theory, briefly stated, proposes that when the condition part of a rule (acting on declarative knowledge embodied in a non-rule form) is satisfied, then the action part of the rule is executed. The conditional part of rules is considered to include goal conditions, providing for the expression of volition or goal through action. Anderson's theory deals with the complexities of both rule and declarative knowledge structures and with a mechanism for conflict resolution to determine the order of rule execution.

Anderson's theory is relevant to analysis because the production (rule) and the declarative knowledge unit (the working memory element) are basic units of subject-matter analysis that have been shown by Anderson's research and development work to be instructionally valid as well as being a useful design and development unit.

Anderson's theory is clear in its distinction of two types of knowledge: the *procedural* and the *declarative*. Less attention has been paid to the declarative knowledge construct, the *working memory element* (WME). However, it must be reckoned as part of the analysis structure, since without it conditions cannot be detected and rules cannot function.

Anderson claims, on the basis of extensive research, three properties for WMEs (which he also refers to as memory "chunks"): (1) an optimal size consisting of three elements; (2) internal relationships that are linearly ordered, spatially ordered, or semantically ordered; and (3) hierarchical ordering relationships with respect to each other. Since there are two types of knowledge element in Anderson's system, analysis output must be expressed in both, though the bulk of Anderson's writing concentrates on the procedural rule form (hence, the title of Anderson's book, *Rules of the Mind*). Critical but understated parts of Anderson's theory are: (1) the emphasis on goals as analyzable structures that act as sequencing elements by entering into the conditional portion of rules during mental processing, and (2) the criticality of declarative knowledge representations to rule functioning.

1.7.5 Clancey

Clancey (1986) provides an interesting and close contrast to Anderson's work. Clancey found that human bodily system states and expert diagnostician intentional goals were not absolutely necessary to the functioning of an expert system for medical diagnosis. That is, he built a rule-based diagnostic program that did not incorporate these things as independent elements of the decision process. The diagnostic expert was capable of achieving correct diagnoses at roughly the accuracy rate of a human diagnostician. But when Clancey attempted to convert the successful diagnostic expert into an instructional expert, it became necessary to add rules to be used in thinking and decision making about performance goals and solving methods. These new goals also incorporated the ability to reason about human body states that made certain rules relevant and others irrelevant at a given moment—body states like "compromised" and "immunosupressed" (Clancey, 1984).

Despite his finding, Clancey continued trying to avoid system process models as a basis for generating explanations. He focussed on expert system knowledge only tangentially, representing the properties and states of a system as parts of the rule condition and not as a separate and interacting coherent body of essential information amenable to analysis. Because of this, it remained difficult to explain to a learner the expert's generation of an action plan. In some cases it forced Clancy to resort to the inclusion of existing textbook clippings as explanatory elements during instruction. That is, the analysis, because it was incomplete, lacked the ability to generate instructional message.

Means and Gott (1988) also found it important to describe the electronic systems that were the objects of decisions by expert system maintainers. This (often informal or secondary) analysis of systems has become a part of the process generally accepted as "cognitive task analysis", but it is not often described as a formal part of the analysis process in its own right.

1.8 State of the Art In Pre-Design Analysis

The current state of the art in pre-design analysis consists of: (1) a relative handful of designers employing methodologies with a strong theoretical basis, (2) a much larger group of designers who practice analysis with methods with a weak theoretical basis, and (3) a majority of designers who use makeshift or no analysis methodology. The instructional technology of analysis, if it is to contribute consistently and reliably to instructional designs, is in need of attention. Particularly, some of the rich theoretical possibilities described in the previous section that are now reserved to researchers need to be opened to average designers in a way that does not require them to earn advanced degrees in analysis. In other words, the technologies that have worked in the laboratory need to be formalized in a way that makes them and their benefits to instruction accessible to designers at large.

The average analyst should be able to answer basic questions about an analysis and its interpretation: “Why am I performing analysis? What am I analyzing? What is the output? What use is made of the output?” Answers to these questions are of much interpretive importance, but more critically, they are required if the analyst is to operate at more than a cookbook level.

1.8.1 PDA in Everyday Practice: Production

The practitioner community for PDA consists of several sub-divisions defined mainly by their economic interests and patterns.

Public educators. At the junior college and high school levels, task analysis is used among public educators mainly to define training requirements in more structured subject-matter areas. Complex technical skills such as paramedical, construction, and information technology are commonly analyzed. Of the more conceptual parts of the curriculum, mathematics is much analyzed, as is science.

Early academic research in analysis tried to bridge analysis practices to a theory base using elementary and high school mathematics subject-matter because of its high degree of structure and its centrality to other curriculum areas (Gagné, 1968; Resnick, Wang, & Kaplan, 1970). Aside from centralized research efforts, individual public educators tend not to use task analysis in their own work (Loughner & Moller, 1998). Individuals find it hard to locate tasks in traditionally organized subject-matter and also lack the time, skill, experience, and incentive to perform what can be difficult and subtle analyses, such as those required for improved reading instruction. Finally, most educators lack the conviction that task analysis will improve their instruction proportional to the effort it requires.

Military training developers. The peacetime military is the world's largest organization for training development. Over the past half century, funding in different categories (research, application, etc.) has been instrumental in creating a systematic development culture and a body of practice for training design that emphasizes standardized procedures. Each service has promulgated its own training development standards that normally include strong emphasis on one or more formulas for conducting and documenting task analysis. Retiring military trainers experienced in this culture of analysis and design are a major source of practitioners for the training development communities described later in this section. Therefore, the military development culture has heavily influenced some areas of industrial practice of PDA.

Government Training Developers. Training developers working within the U.S. government, likewise, normally work under the requirement of organizationally mandated formal instructional development processes. However, enforcement of these standards—which normally include a requirement for some form of PDA—is not as pervasive and strict as in the military. There is, therefore, a greater variety in analysis and development practice.

Commercial Training Developers. Large commercial contractors provide commercial training packages and training development services for industry and government. The largest single sector of this training is in the information technology area (hardware and software skills training). Frequently a form of PDA is performed during this development that breaks the subject-matter into task-centered lessons. Many of the companies that offer commercial development services are staffed heavily with exmilitary developers. These workers bring with them the emphasis on formal PDA that they learned in the military.

Corporate Training Developers: Large Scale. The community of corporate training developers falls into two divisions: (1) large-scale developers, and (2) small-scale developers. The large-scale developer group consists of corporate units who fill large training contracts in service to their company's products or hardware contracts. An airplane manufacturer can support large-scale, high-end training development efforts with the expectation that customers will see sophisticated training packages as value-added to the aircraft product. Likewise, most electronic and hardware-software systems contractors maintain large training development staffs. These large organizations frequently have their own design process, including, in virtually every case, a prescribed PDA procedure. Such organizations are often willing to adopt (or adapt) to the specification of a prospective client, such as the military or government.

Corporate Training Developers: Small Project. Small-project corporate training developers consist of corporate employees located within a company's training development organization. Their assignment is to supply internal corporate training in the form of relatively small and often short-lived products through individual development projects. Skills to be trained are often proprietary to the company and may be seen as a competitive factor by the company. Formal instructional development process in this community of developers is irregular. Many corporate organizations that have training design and analysis standards are not successful in mandating those standards to individual projects. Many organizations of this type have no development standard or have only a loosely defined standard. In many cases, a relaxed standard is adopted to train what are perceived as “soft” or fuzzily defined skills.

Designers working within organizations of this type often lack formal training. What trained designers these groups do have often come from academic programs rather than from the military. Designers in this group, therefore, tend to have less experience and interest in conducting formal analysis. When analysis is conducted it is frequently on a topic basis or in accordance with a corporate doctrine of instructional objectives. Such less formal analyses allow subject-matter experts—who are normally supplied by a within-company client—to participate in the development process more comfortably and without learning new and threatening technical methods that would give the designer greater control over the product than the client.

Summary. In section 1.3 we discussed the disincentives to analysis due to job assignments. Here we have looked at incentives from the organizational perspective and seen that some (military, government, large-scale contractors) have good reasons for attention to analysis while others (public educators, small project developers) have what they feel are good reasons to ignore or oppose it.

Among those who do practice PDA, there is reason to suspect great variability in process, terminology, and product utility, even among those who are working according to a standard process and product specification. In all of the organizations it is more typical that the practitioner lacks formal training in instructional design and analysis. This means that there currently exists an army of instructional designers who have arrived in their positions from other disciplines (technical writing, HRD, star performers, promotion/demotion or technical development such as graphics and programming). These people normally know little about PDA or its basis in theory.

When one performs PDA he/she is entering a highly specialized world full of difficult skills, varied terminologies, and uncertain outcomes—most often doing so with little formal preparation. Rather than wondering why a such a small percentage of designers perform PDA, we might wonder at the large amount of effort it took to instill the relatively spare standards that do exist today, producing some cultures that do perform PDA routinely.

1.8.2 PDA in Everyday Practice: Maintenance

Maintenance of analysis is also an important issue. A main purpose of analysis mentioned earlier is *requirements tracing*—the linking of systematically derived training needs with specific elements of a training solution (Jarke, 1998). Once linkages are formed, changes automatically break them: changes to procedures, changes in policy, changes in equipment, changes in personnel roles. During renovation and extension of a major building on a university campus, it was found that electrical blueprints had been lost. This resulted in a work delay of nearly a year as the circuit connections of electric wires left exposed by wall removal were traced. Losing the analytic component of a complex instructional product's blueprints has a similar effect.

It seems clear that some design projects do not have sufficient scope and risk to merit the time and effort of formal analysis. Where size or risk factor is high, however, both initial analysis and analysis maintenance are advised. In certain high-risk areas, it is mandated (e.g., aviation, nuclear power). PDA methodologists must develop decision rules to help designers judge whether or to what extent analysis should be pursued. Once analysis is indicated by those rules, it should be maintained, along with its links to media elements of the instructional product.

Consider the plight of the average designer given the assignment to revise and update an existing instructional product whose design was originally based on PDA. For this example, let us assume that traditional task analysis was performed and that the designer must make a few significant changes to the product. Changes to the media portion of the product begin with analysis documentation:

- The designer must learn the internal logic, assumptions, and structural metaphor of the original analysis.
- The designer must detect the consistent patterns of linkages between analysis and instructional product and determine whether they were consistently applied.
- The designer must re-enter the analysis revising, adding to, and restructuring, as necessary.
- The designer must trace the implications of changes forward to all parts of the instructional product.
- The designer must make changes to the instructional product that are consistent with the analysis-to-product linkages of the original product.

These steps present a major challenge from the beginning because, in many cases where analysis has been performed, it is not kept current. This produces the result that it no longer traces forward to elements of the instructional product. The additional burdens of time, expertise, and the long chain of linkages between analysis element and design element only reduce the likelihood that analysis will be a factor during product revision.

1.9 Theoretical Foundations of PDA

To clarify the theoretical foundation for PDA, questions must be raised concerning the nature of the inputs and outputs of an analysis, the nature of the transformations worked by analysis, and the manner in which the analysis product links to later instructional use. In this section we discuss these issues.

1.9.1 The Problem of "Knowledge"

We challenge the accepted idea that an analysis captures "knowledge" as output. We take the position that what is captured cannot be viewed as knowledge in the classical sense and that doing so restricts our vision of possible analysis methodologies and applications. Traditional task analysis does not make claims that its output represents knowledge, but knowledge capture, knowledge representation, and knowledge analysis are consistent claims of CTA. A complete literature has developed about knowledge capture, knowledge storage, knowledge representation in storage, knowledge representation at the human interface, and incremental knowledge acquisition. For decades conferences have been held and books have been published (Brachman & Levesque, 1985) on the subject of "knowledge" and its appropriate representation for both instruction and the construction of human-like reasoning machines.

The popularity of the term "knowledge" in this modern usage derives from early artificial intelligence research where the term was used almost as a kind of hypothesis. Use continues today because "knowledge" has become a paradigmatic term—though it is still undefined. The specialized sub-group of AI researchers who build intelligent tutoring systems (ITS) have been persistent users of this terminology, pursuing, among other things, the problem of uniting learning theory with instructional theory. For this group the assumption of knowledge representation is an important part of theoretical arguments.

In its early context of use, the term "knowledge" represented a body of expert rules and concepts structured according to some organizing principle that could be captured, embodied in a computer program, and made to perform human-like or expert actions or judgment. It has become apparent in the work of Clancey (1985) and in such projects as Big Blue, the chess-playing computer program, that the expert rules used by computers to perform human-like feats can be leveraged by computing power and do not necessarily use the human expert's rules. This raises questions about the validity of the knowledge representation argument. Even theorists like John R. Anderson who are careful to base their knowledge-capturing rule and memory element systems on human protocols and strict theoretical guidelines do so while reminding us that they are still dealing with only limited portions of the phenomenon of human knowledge. In its living form, this phenomenon intertwines fact and principle with emotion, value, personal experience, memory, and self-consciousness. Over time, as ITS researchers have adopted the principle of modeling domain knowledge and using corresponding or overlaying models of student knowledge (the "student model"), the implicit assumption has become that this constituted capturing and representing "knowledge".

The trend in "knowledge" studies has been heavily influenced by theories of mental models, schemata, and production rule systems. All of these represent a computable form of "knowledge". These forms are intended to support generation of all or part of the message elements and interactions during instruction (and, since generative instruction is simply real-time design, during non-computerized design also).

Though this research has produced provocative instructional ideas and a variety of representational schemes, the term "knowledge" has never acquired a definition that is convincing in scope and general applicability. Some research (Anderson, 1998) has shown results in limited and highly-structured subject matters. Other research (Schank, 1994) has shown application to a broader range of subject matters but has had to sacrifice much computability to do so, and extensive conditioning steps are required following analysis to prepare instructional message and interaction. This problem continues to create, for instructional designers, a sense that "knowledge" exists in some externalizable form without giving enough definition so that a designer can use the concept as a general purpose productivity tool.

1.9.2 Analysis and Automated Design Systems

The problem of the analysis structure in relation to the design structure becomes apparent when attempts to automate the instructional design process are considered. Automating design systems—a project with enormous economic implications—forces the designer to be explicit about elements of design, elements of content, elements of instructional message, and elements of instructional logic to an extent never required previously.

We believe that the examples above highlight the importance of the nature of the structures used in design. Most designers do their work without being self-conscious about the properties of the elements they draw together into a design.

1.9.2.1 Merrill and Drake—Simulation Design Systems

Instructional Transaction Theory (ITT) described by Merrill and others, (Merrill, Li, & Jones, 1991) rely on the capture of taxonomic types of knowledge and its representation to students through expert system mechanisms acting upon a "knowledge base". Knowledge is represented in the form of a PEAnet, or a "process, entity, activity network," Merrill, et al. define a PEAnet as the set of interrelationships "among processes, entities, and activities [which] enables the construction of learning environments from knowledge objects (p. 412)." The knowledge question is thus directly relevant to Merrill's attempts to automate instruction derived from the PEAnet and given manifestation using stored resources through transaction algorithms and message construction algorithms.

Drake (1997), building on Merrill's work, demonstrated a version of this principle in a simulation design program that used self-articulating "knowledge objects" to construct a variety of presentation, demonstration, and practice messages and interactions automatically from primitives supplied by a subject-matter expert working in conjunction with a designer. This design team working together created a mini-language for instructional dialogue that can use the terms of the structured content language as objects to be manipulated by instruction-forming rules expressing themselves through slotted messages into which the terms of the content language are substituted.

The computational techniques used in this approach to automated instructional (real-time) design echo techniques used by ITS designers over two decades of research (Wenger, 1987). The important question raised by Drake's research is whether the objects that lie at the center of Drake's system represent "knowledge" or merely dynamic artifacts capable of entering into standard instructional computations. We argue the latter. Moreover, we feel that Drake's system illustrates the principle that design and analysis hold mutual influence with one another stronger than most designers understand.

Drake's simulation builder—probably misnamed since it also builds instructional message and interaction from primitives—presupposes in its instructional design that only certain kinds of primitive construct will be supplied to it. It assumes that the input will be in the form of simulation model components and their properties (compare this with Lesgold's object analysis) as well as a set of interrelating rules. In addition, Drake's system assumes the input of: (1) key words associated with components, and (2) key representations that provide visualization of model components in their possible states. These representations serve the same function in the construction of templated visual communications as the keywords serve in the construction of templated verbal communications.

The key issue illustrated by Drake's work is that it requires as an output of analysis raw elements that are not supplied by any of the widely-used task analysis methods and that are imperfectly supplied by cognitive task analysis as it is currently described. It is capable of producing instructional visuals, text, and strategic interaction, but it is only capable of doing so because it extends existing analysis methodology.

1.9.2.2 Monolithic Design Systems

Designer's Edge ™ (Allen Communications, 1994) is an automated instructional design product based upon the principle that what is created during instructional design is a database with elements that chain forward through several transformations and linkages to an instructional media product as shown in Figure 3 (Gibbons & O'Neal, 1989). *Designer's Edge* leads a designer through PDA which links forward through several steps to product, message, and computer logic designs for the creation of computer-based instruction. The end product of this forward linking process—specifications for computer logic and pre-composed message elements—can be "poured-over" into multiple computer-based instruction authoring tools.

A system of this type confronts the instructional theorist with the issue of "Where is the knowledge?". *Designer's Edge* makes no claim about knowledge. A designer can create a complete instructional product without becoming aware that an intermediate representation of "knowledge" is a factor. Designers approach their task from the perspective of creating "artifacts" as opposed to "knowledge."

1.9.3 Design Constructs

Most structures used in design are described by design theorists, along with rules for applying them during design. So it is the responsibility of the theorist to build the rationale for a prescribed set of structures. That rationale normally traces back to the nature of structures derived through pre-design analysis and their interpretation. Different design theories rest on different theories of structure.

Gibbons and Lester (1996) describe four conceptions of instructional product. Each is defined in terms of the structural constructs valued and given priority by the designer:

- The **media-centric** view of instructional product describes instructional force in terms of the selection and arrangement of media constructs—sensory properties, layouts, displays, changes in display properties, etc.
- The **message-centric** view of product describes the product in terms of the selection and arrangement of message or information elements—organizing ideas, subordinate ideas, explanations, mnemonic arrangements, demonstrations, "interactivity", etc.
- The **strategy-centric** view of product holds that product effectiveness arises from the selection and arrangement of message and interaction elements according to instructional formulas or strategies—expositions, taskings,

message sequence, feedback, prompting, task ordering, etc.

- The **model-centric** view connects product strength with its ability to supply learner exposure to and interaction with appropriately chosen media-independent models of environments, cause-effect systems, and expert performance.

Early *media-centric* research compared forms of media expression (film, printed word, drawing, photo, etc.) for their ability to instruct (Lumsdaine & May, 1965). In addition to searching for the principles to guide media assignment, researchers tried to identify uniqueness in the instructional capabilities of different media, hoping to find rules for the selection of the right medium for a particular type of instructional goal. *Message-centric* research sought the principles for designing message structures. Ausubel (1968) studied the effects of advance organizers on the learning. Wittrock (1974) and Rothkopf (1996) proposed principles for organizing messages and for designing interactions to enhance message uptake as parts of their theories. *Strategy-centric* research seeks effective patterns of standard message elements, patterns of response-seeking and judging, and ancillary support that lead most efficiently to learning.

A current trend in instructional theory toward *model-centered* design structures is typified by this statement of Montague (1988):

> *The primary idea is that the instructional environment must represent to the learner the context of the environment in which what is learned will be or could be used. Knowledge learned will then be appropriate for use and students learn to think and act in appropriate ways. Transfer should be direct and strong.*
>
> *The design of the learning environments thus may include clever*

combinations of various means for representing tasks and information to students, for eliciting appropriate thought and planning to carry out actions, for assessing errors in thought and planning and correcting them. I take the view that the task of the designer of instruction is to provide the student with the necessary tools and conditions for learning. That is to say, the student needs to learn the appropriate language and concepts to use to understand situations in which what is learned is used and how to operate in them. She or he needs to know a multitude of a proper facts and when and how to use them. Then, the student needs to learn how to put the information, facts, situations, and performance-skill together in appropriate contexts. This performance- or use-orientation is meant to contrast with formal, topic-oriented teaching that focuses on formal, general knowledge and skills abstracted from their uses and taught as isolated topics. Performance- or use-orientation in teaching embeds the knowledge and skills to be learned in functional context of their use. This is not a trivial distinction. It has serious implications for the kind of learning that takes place, and how to make it happen. (p. 125-6).

The four stages of design thinking we have described form a general sequence through which individual designers tend to move, just as have research trends and general design practice. As the current flood of new multimedia and web technology users enters professional instructional design, many are being told that the power of instruction lies in the technology itself. Therefore, a whole new and very large generation of media-centric designers is coming into first contact with the structural design elements called “page”, “frame”, and “resource”, depending on these elements to lead to effective instruction. We can expect a maturation of this group over time as they accumulate experience and observe their products at work with students. As they do so, they will inevitably experience a shift in the primitive constructs they use while thinking about designs.

We, however, favor the model-centric view of instruction, one in which the "model" the "instructional event" and the "instructional problem" are appropriate central constructs of design. From these model-centered constructs can be derived media, message, and strategic constructs that are the secondary, not the primary, constructs of product design. These constructs are identified through methods of pre-design analysis appropriate for capturing them. A key point is that the analysis process can be adjusted by modifying its inputs, outputs, and transformation rules (See Table 1) to produce the type of output construct the designer desires. What is considered useful by a designer depends largely on what he or she has accepted as design assumptions. This refers not to the principles of instruction but to the principles the designer will use for generating instruction, a set of options that are explained in the next section by a discussion of the projection principle of instruction.

1.9.4 The Projection Principle

Projection is the process of bringing information to a display surface. All instructional systems involve projection from some source to a display surface. Instruction is the process of transforming one or more sources of message and interaction into a projection. We speak of a computer's display as an interfacing surface between CBI and the learner. In a more general sense, if we define all of the coordinated senses of the learner as the display surface, then all instruction projects representations with some degree of synchrony onto this surface. Interactive instruction also synchronizes responding through this surface back to the source.

An average computer-based instruction display contains fragments of verbal content message, graphic content message, verbal directions, controls for mouse responding, selection areas and content on menus, and perhaps other represented display elements projected onto a single surface that looks to the learner as if they came from a single source. Regardless of the number and nature of sources from which these displayed elements arise, they are projected by some form of display management function that is responsible for bringing them to the display surface in appropriate synchrony. One of the main services provided to a designer by CBI authoring systems without the price of complex programming, is this built-in display management.

A live instructor, likewise, combines elements from diverse message sources into a coherent message package displayed in synchrony. Analysis of live instructor's communications reveals the same kinds of elements combined there that might be combined in computer-based instruction. However, the live instructor's range of display modalities differs from the computer's. But a live instructor's words, gestures, presentation of overheads, drawings on the blackboard, and other elements of message can still be seen as parts of a synchronized projection. Some elements of this projection arise within the instructor from decisions about the content, while others arise from decisions about managing the event of instruction or the actions of the learner. The totality of these message elements is managed and timed by the instructor's own display management routines that constitute what we normally think of as the instructor's technique or style of presentation.

The origin and generation of the message elements that are translated into expressions or representations—whether by a live instructor or by an automated instructional product—is of great importance to the instructional designer. Instructional theorists seek general principles to govern and mechanisms that accomplish message generation and projection.

If we reverse engineer different examples of computer-based instruction (for ease of comparison) we will see different sourcing patterns for message generation, and that will in turn yield implications for PDA.

1.9.4.1 Projection From Pre-Composed Frames

Some forms of CBI project displays from pre-composed message-and-logic bundles consisting of pre-written and stored text messages and pre-drawn visual resources (See Figure 4).

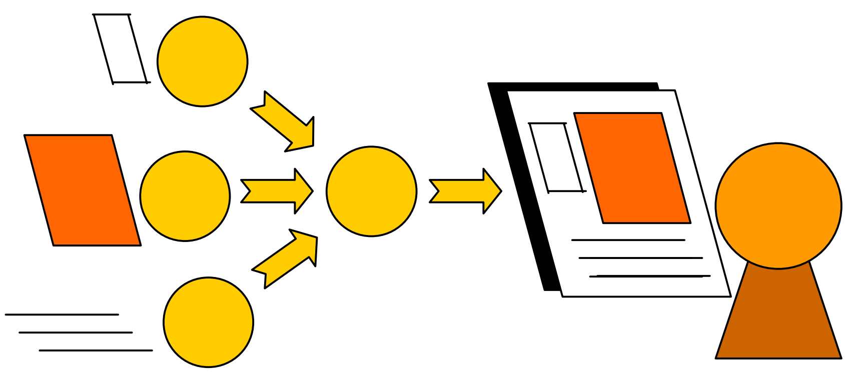

Figure 4. Static projection from pre-composed resources.

These bundled elements of message are often placed into logic-interconnected sequences authoring system “frames” or “pages”. Logic interconnections can be as simple as ordered chaining or can include branching. With this organization, it appears to the learner that displays are being projected forward from a single source. The generative mechanism of the designer's thinking has already accomplished projection from other, more primitive, elements of message and interaction structure and from multiple sources within the designer's mind onto the authoring plan. Therefore, frames themselves are an intermediate projection from a designer's internal sources through a design process not normally open to inspection.

1.9.4.2 Projection of Database Elements

A slightly more complex form of CBI draws pre-composed elements (text files, graphics files, audio files, etc.) together at run-time from libraries of multimedia resources according to some formula: a computation, a set of rules, or the records of a database (See Figure 5). Products using this mechanism have a greater degree of flexibility because some of the resource selection and display management decisions can be computed at run-time.

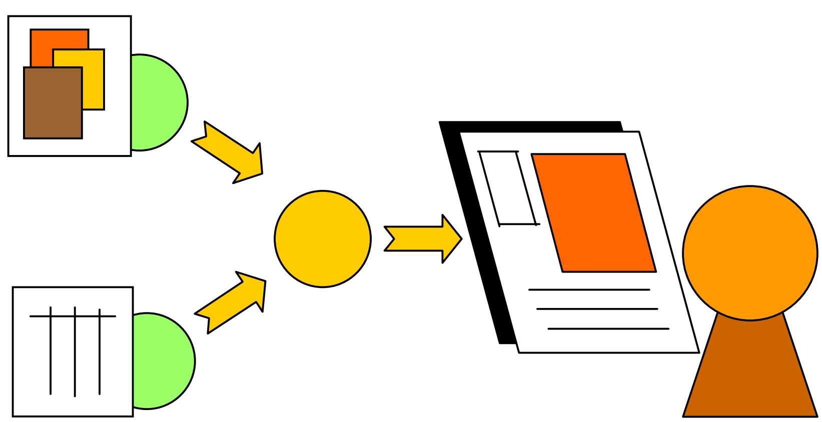

Figure 5. Dynamic projection from pre-composed resources

1.9.4.3 Real-Time Generation of Displays

An important issue of intelligent tutoring systems (ITS) research has been the generation of instructional displays entirely from primitive elements, which eliminates the need for designers to pre-compose displays or even to database message elements. Message composition in an ITS is computed from models of the content, the instructor, the student, and the expert performer, or from some combination of these (See Figure 6), often using primitive message elements and slotted message templates. Display management, since display content is not known in advance, must be computed at the time of instruction, and often slotted screen templates or a set of rules of display composition are employed.

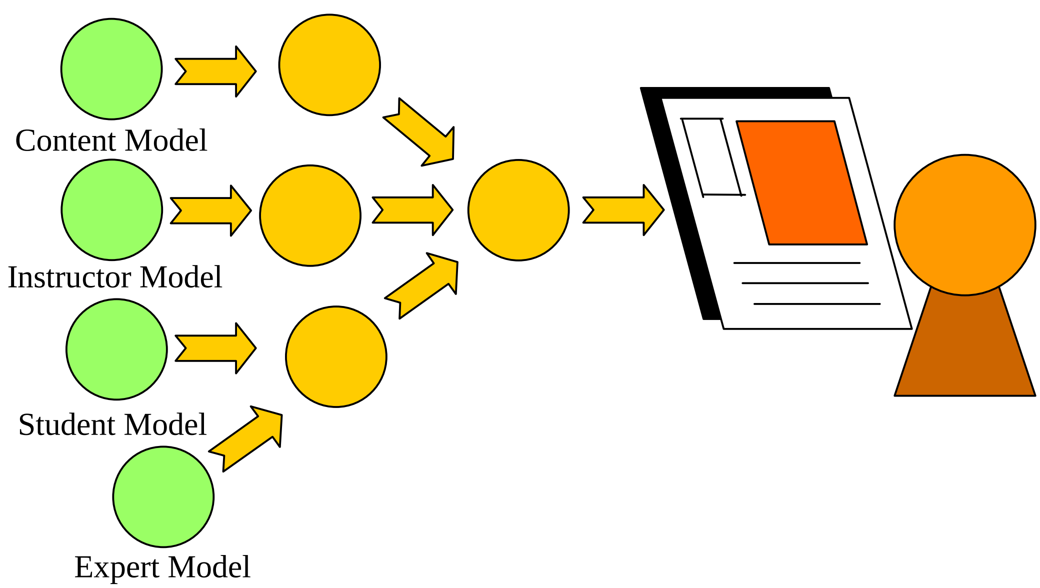

Figure 6. Dynamic projection from primitive elements

1.9.4.4 Examples of Projection

Today's ITS researchers continue to pursue a problem introduced 35 years ago—machine design of instructional message and interaction. Enough success has been attained in solving this problem that it is possible to contemplate machine-generated instruction that approximates or duplicates some forms of designer-generated instruction. One of the keys to increasing machine generation of instructional message and interaction is PDA.

The many combinations of sources that can be used as the basis for the generation and the projection of instructional message have been insufficiently explored by design theorists. Mitchell Resnick of MIT explores one alternative with his "Active Essay" series published on the web (Resnick & Silverman, 1997). In this series, Resnick divides the computer display into two areas, each fed from a different *type* of source.

In his essay "Exploring Emergence", for instance, the left, textual, portion of the display is supplied from a database of pre-composed and stored textual messages. The right, visual, portion of the display is supplied from a Java applet that models a complex process. This applet is fed initial set-up values from a database at the initialization of each "page" of the sequenced essay. From that point the learner is free to change set-up values and use controls for starting, stopping, and single-stepping the visual model (in this case of *The Game of Life*) produced by the applet. When the "page" of the essay is turned, this cycle of left-text, right-applet-with-initial-values repeats. This example highlights the extent to which we have allowed our conceptions of display sourcing to be limited by our design categories for the architecture of products—page-turning tutorial, databased tutorial, and simulation.

1.9.4.5 Implications for PDA

But exploring the possibilities for mixed projection sourcing implies that designers will use methods of analysis appropriate to each variety of source and that the analysis process itself will be compatible with multiple sourcing types—some of which may call for pre-written, pre-drawn messages, and some of which will be generated at the time of instruction. The current state of the art in analysis provides no such tools. Especially, it is deficient in supporting the definition of problems and problem sequences from which can be derived the primitives to support mixed-source projections.

What had to be captured during PDA in order to make Resnick's product possible? In order to create the right side of the display it was necessary to create a model engine and multiple sets of rules for model operation, as well as initial values for different set-ups. For the left side of the display the PDA must have identified a sequence of "maskings" of the model described for the right hand side. The function of these maskings is to reveal the right hand side model incrementally, through a series of simple explanations followed by a set of activity directions. The net effect of the sequence of maskings is to reveal, piece by piece, the complicated model underlying the right hand side dynamics. This approach to instruction illustrates model-centered principles like carefully sequenced problems that allow the learner to interact with appropriately denatured or masked models (Gibbons, 1998). Additionally, this is an example of how the appropriate analysis resulted in a model-centered sequence of "problems" or "posings" that Resnick could relate to maskings of the model.

Neither the model nor the problems used to create maskings can be correctly characterized as "knowledge". They are appropriately viewed as experience-producing artifacts generated based on some analytic unit: the problem. We feel that the major challenge of PDA in the future will be the selection of appropriate models for which to perform analysis. The analysis will then describe the environment of the models, the systems that embody the models and make them available for learning through manipulation, and the expert performances associated with the models.

1.10 Practical Analysis Dynamics

In addition to the theoretical issues just discussed , there are several practical issues that act as constraints on the development of PDA methodology. Some of these practical issues arise from misconceptions of the nature of the analysis output, from the characteristics of human personal knowledge, the energy and intensity required to perform analysis, and the necessity of understanding in advance the fit between analysis output and design input.

1.10.1 The "Truth" Factor

Performing a PDA is not a search for absolutes or a search for "truth". PDA is a process of identifying *satisficing* solutions. While performing a traditional task analysis, an analyst invents phrases that represent tasks. The specific phrasing of tasks depends on the analyst's viewpoint and experience interacting with the knowledge of the subject-matter expert (SME). The boundaries of tasks, the stopping and starting points, the extents, the conditions included in them, all are products of the analyst's judgment. Therefore, a PDA methodology should not strive to represent "truth" (things as they *really* are) as much as it strives to create useful representations of things and phenomena that allow learners to learn to act productively and precisely within real world situations by interacting with denatured representations of them. Millitello and Hutton (1998) describe how differences in terminology, in world view, and in levels of expertise, all influence the representation that results from CTA. The output of analysis should not be judged in terms of conformance with an abstract "truth" as much as they should be compared for whether they capture the essence of environments, systems, and performances that a learner must negotiate. Analytic thinking during PDA is a technological activity. In seeking a *satisfactory* and *useful* solution, multiple answers may be found.

One of the important results of analysis is the building of the system of categories and terminologies that are most useful to performers. Lesgold (1999) suggests departing from the traditional assumptions of analysis that lead us to capture the "real world" as it exists in its full complexity. He has discovered that it is, in many cases, desirable to describe the world to the learner at a level of simplification that still enables the learner to remain functional in real-world situations. The discovery was forced upon Lesgold by repeated experiences with systems that have become so complex, particularly in the world of computers, that few humans fully understand system principles in complete detail. Lesgold's finding is very important to PDA designers because it requires them to form consistent principles for simplification of complex systems. Lesgold's findings echo the earlier findings of Kieras (1988).

1.10.2 Implicit versus Explicit knowledge

Research indicates that a portion of human knowledge is appropriately called *implicit*, meaning that it is not held by its possessor at a level of conscious recognition. Implicit knowledge appears to be common, especially in everyday routine tasks. The special problem of implicit knowledge (and much "expert" knowledge seems to be implicit) is that there are no terms by which its possessor can refer to it.

The constant practical problem during analysis with subject-matter experts is that SMEs often do not know how much they know and do not realize how much knowledge and skill they actually bring to the problem. They also tend to be oblivious to the shortcuts in their own thinking—particularly the manner in which they simplify complexities—that allow them to address problems efficiently and with correct results (Chi, Glaser, & Farr, 1988).

Analysis involving an analyst and a SME can be viewed as an oppositional process. The analyst's function includes questioning, doubting, obtaining insight, naming, and exploring the knowledge of the SME, assisting the expert to draw forth expressions that are explicit. When both analyst and SME understand this dynamic of their relationship and the purpose of the oppositional process, then the productivity and utility of the process increases.

The implication for the analysis methodology is that it must help the SME to recognize and analyze pockets of implicit knowledge. This probably includes capturing the categories and terms that already exist in SMEs thinking, using them as stepping stones to more accurate and complete descriptions. It must do so in a way that does not demean the SME, reduce SME confidence, or threaten SME position of expertise.

1.10.3 Fatigue factor

Analysis is an unusually intense activity that resembles, in some ways, a dialogue between a prosecuting attorney and a defendant. Because of its intensity, analysis rapidly uses up the energies of those who participate in it. Frequently after one half day of intensive analysis activity, both analyst and SME find it difficult to maintain intensity. The implication for analysis is that methodology must be sensitive to the sources of stress that drain energy and must seek ways to achieve greater efficiency. One route to this goal may be to do away with the analyst's terminologies and orthodoxies in the relationship with the SME. Analysis, should to the greatest extent possible, take advantage of what the SME knows, allowing the SME to express it in terms compatible with daily practice. The role of the analyst can then become that of a cooperative reviewer more and an adversary less. The familiar terms of the process should help the analysis team move ahead, making as rapid progress as possible, rather than focusing attention onto the orthodoxies of the analysis technique itself. The process should place into the hands of the analyst the questions to ask and the means of rapidly recording the answers that will move things ahead expeditiously.

1.10.4 Vision of Outcome and Its Application

Frequently SMEs drafted for analysis duty have no idea what the outcome of analysis will be and have no vision of how the output of analysis will be applied to instruction (though SMEs often have in mind what they think instruction should be like). Cognitive task analysis is a step (Means & Gott, 1988) toward bringing the terminology and logic of analysis into the SMEs world rather than forcing the SME into the analyst's world.

There still remains the problem, however, of giving the SME a vision of the relationship between analysis product and instructional product. The implication for new analysis methodologies is that they must somehow provide this vision, enabling the SME to exercise an even greater degree of judgement and selectivity and enabling a greater portion of the work to be done by the SME independently of the analyst. Joining the output of analysis as closely as possible with the characteristics of design and ensuring that designs are expressed in familiar terms are part of that solution.

1.11 New Instructional Paradigms

The solution to the PDA problem has theoretical and practical roots, but is in reality only one part of a larger problem: the problem of a shifting instructional paradigm that has resulted from over twenty-five years of cognitive research and theorizing. The new paradigm is represented by:

- Increased interest in the sociological setting in which learning takes place (learning within contexts like those in which the learning will later be used).
- Interest in levels of knowledge beyond the traditional domain knowledge (learning problem solving patterns and heuristics and learning to self-instruct).
- New concepts for sequencing instructional experiences.
- New methods of instruction that stress problem solving as the occasion of learning, learning by doing, learning from coaching and feedback, learning by articulation and reflection.

1.11.1 Cognitive Apprenticeship

These trends are summarized in the principles of an instructional theory called Cognitive Apprenticeship (Collins, Brown, & Newman, 1989). A review of multiple examples of new-paradigm instructional products that span a broad spectrum of differences can be found in Gibbons and Fairweather (2000). The stimulus for a new analysis technology comes in part from the encroachment of this new view of instruction. It does not replace more traditional views of instruction but assimilates them into a larger framework of instruction.

The focus of cognitive apprenticeship is the approximation toward expert behavior of an apprentice through observation and imitation in a supportive environment that incorporates coaching, scaffolding, and reflection. The cognitive apprenticeship theory expresses an operational principle for the design of instructional systems from which an enormous range of individual configurations can be derived. These are the examples reviewed by Gibbons and Fairweather (2000).

1.11.2 Problem-Based Learning

A second theory of instruction titled Problem-Based Learning (PBL) is central to the paradigm shift. It is based on the requirement that students solve problems to learn and that solving take place in an environment containing special provisions for support of learning. PBL also emphasizes to a greater extent the non-expert modeling of knowledge and its construction by learners in the absence of expert behavior to observe.

1.11.3 Model-Centered Instruction

The theory of Model-Centered Instruction already discussed also qualifies as new paradigm. It is intended as an instructional design theory that sees the instructional design process from the designer's view and attempts to articulate conceptual structures unique to the needs of the designer. Furthermore, MCI places greater emphasis on the instructional means to support the learning of non-expert (natural and fabricated) systems and environments.

Taken together, these three emerging instructional theories, define instructional methods that provide a new set of essential constructs for instruction. They de-emphasize some design structures that have become familiar through usage: sequenced instructional message and structured instructional strategy. As a new central organizing structure for instruction they provide the problem, the problem solving environment, and the models to be learned. This experience is carried out in such a way that the solving of the problem with support instructs and supports student learning of new skills or concepts.

This shifted emphasis to a new central organizing structure—the problem—to a great extent calls into question existing instructional design methodologies whose logic tends to lead to strategy- and message-centered designs. We need analysis methodologies that identify models and problems.

1.12 Analysis/Design Languages

In order to facilitate the design of instructional products whose central construct is the problem and the model, it is useful to conceive an analysis methodology where problems and models are the output. This aligns the analysis and design methodologies in a way that has not been true of traditional instructional design processes. But, it is only one example of an approach to design that is becoming general in other fields. For example, in the field of computer programming, the Universal Modeling Language (UML) has evolved over the past decade in response to the paradigm shift in programming toward object-oriented programming (Booch, Rumbaugh, & Jacobsen, 1999).

The paradigm shift to programming objects has created a new central construct for computer programs—the object. Traditional programming practice has relied upon the code sequence and the sub-routine as analytical units and the "top down" and "procedural" program design approaches took advantage of that world view. Programming with objects does not readily yield to the same kinds of analysis previously used. The shift in the programming paradigm has occasioned the shift to combined analysis and design languages in which the terms of the analysis language match the terms of the design language. This shift allows the designer to analyze and design at the same time by building structures that consist of self-contained, atomistic units.

UML is a notational scheme that joins symbols together in a syntactical diagram framework. Symbols are assigned constant semantic meaning, and notation is disciplined. “Views” of a system can be represented in this semantic/syntactic notation in such a way that the meaning of the view to one programmer will be essentially the same as its meaning to another programmer. Multiple views on a system are created which answer specific questions about aspects of the architecture, connectivity, logistics, operation, and structure of the system at several levels of detail.

UML is a language for the design of object-oriented programs, but it is also a language of analysis. With it, a programmer can break down and capture complex system elements that have direct application to designs. Some of the principles of UML that appear to apply to the PDA problem are:

- The definition of a standard set of syntactical rules.
- The definition of specific semantic classes of objects.
- The ability to view a system constructed of these objects and relationships from different perspectives and at different levels of detail.
- The standard representation of artifactual structures.

UML gives designers a standard set of conceptual building blocks for design. The creation of UML-type design languages that combine the unit of analysis as the unit of design will not be confined to the programming field. We believe that this analysis/design approach is highly appropriate and desirable for application in the field of instructional analysis and design.

1.13 Required Functions and Features of a Robust PDA Methodology

Having examined at length the conditions and methods of pre-design analysis, we reaffirm the conviction that pre-design analysis is a foundation-building critical step in the chain of systematic instructional design processes. We have pointed out numerous factors having either practical or theoretical impact on the design of methods for pre-design analysis. In the companion white paper to this one (Gibbons & Nelson, 1999) we attempt a redefinition of analysis that meets new expectations, the most important being the non-traditional bridging of the analysis to design constructs—in this case problem structures.

The following is a list of the fuctions and features that we believe an effective PDA should include as discussed in the preceding section of this white paper.

An effective PDA methodology:

1. Provides the substance of goals.
2. Focuses instructional events.
3. Provides structure to instructional environments.
4. Provides the structural basis for the construction of performance benchmarks.
5. Supports generativity: the ability to generate instructional message and interaction.
6. Links design and development processes.
7. Provides logic and quality understandable by those outside the analyst/designer audience, such as administrators and managers.
8. Provides predictable time-skill requirements for administrators/managers to use.
9. Provides predictable output.

10. Contains ability to measure quality.

11. Contains ability to measure completeness.

12. Allows for moderated skill/experience level for analysts.

13. Allows accessibility to non-degreed designers/developers and to SMEs.

14. Provides analysis of risk-intensive subject matters.

15. Removes SME threats from

    15.1. Exposure of lack of knowledge.

    15.2. Loss of leadership role/control.

    15.3. Time consumption.

    15.4. Lack of knowledge of analysis use.

    15.5. Lack of confidence in process, utility of process, value of process.

16. Capitalizes on existing SME views of the world (content, terminologies, etc.).

17. Fits into reasonable/comfortable working patterns of analyst and SME.

18. Reduces stylistics of analysts.

19. Gives consistency of output.

20. Provides ability to measure progress.

21. Provides recording tools capable of adequate volume at speed.

22. Provides tools that build analysis output forward to design elements.

23. Provides tools that represent ideas in their relational forms.

24. Provides tools that match a variety of patterns and formats.

25. Provides tools that support maintenance of the analysis.

26. Analyzes systems, environments, and expert behavior to the level of need only.

27. Brings ITS to real designers.

28. Provides useability by an army of non-degreed ID draftees.

29. Provides maintainability.

30. Supports range of analyses for mixed-source projections.

31. Supports SME trying to articulate “implicit” knowledge.

32. Reduces stress causing factors from

    32.1. SME-Analyst Opposition/Controversy.

    32.2. Analyst arrogance.

33. Facilitates and even suggests sequences of problems for use.

34. Maximizes SME-familiar and SME-comfortable activities.

35. Facilitates vision of SME toward instructional product/results.

36. Supports independent use by SME where possible.

37. Identifies models and problems.

38. Makes use of design language-type representation systems.

39. Provides requirements tracing.

40. Provides focus/scoping statements for compartmentalization of events.

41. Provides primitives for design.

42. Ensures completeness.

43. Ensures minimum sufficiency.

44. Provides engine-like mechanism operating at successive levels.

45. Defines inputs/transformation/outputs.

46. Ensures identical output at all levels.

47. Provides consistent transformation rules.

48. Supports development of new-paradigm instructional products.

In addition to laying out these related functions and features we present a theoretical basis for pre-analysis to discipline and guide the design of other analytic methods:

- Analysis is a technological process governed by the rules of utility and artifact construction.
- The process of analysis begins by identifying classes of conceptual artifacts to be analyzed and by identifying one artifact in each class to serve as a starting point.
- Analysis continues by consistently applying a set of transformation rules to the input artifact and to all artifacts generated until a stopping point is reached.

These propositions describe an operational principle for analysis: the operation of information on information to produce new information. We believe that this principle defines a range of alternative analytic approaches and supplies an evaluative standard for them. We look forward to new developments in analytic technique from this theoretic view and a subsequent expansion of accessible tools for use by everyday instructional designers.

References

ACM Turing Award Presented to Jim Gray of Miscrosoft Research. (1999). *ACM MembeNet (Supplement to the Communications of teh ACM), 42*(6), 1, 13-14.

Allen Communications. (1994). Designer's Edge™. Salt Lake City, UT.

Anderson, J. R. (1993). *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Lebiere, C. J. (1998). *The Atomic Components of Thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Ausubel, D. P. (1968). *Educational Psychology: A Cognitive View*. New York: Holt.

Banathy, B. H. (1987). Instructional Systems Design. In R. M. Gagné (Ed.), *Instructional Technology: Foundations*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Bloom, B. S. (Ed.). (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals*. New York: David McKay Company, Inc.

Booch, G., Rumbaugh, J., & Jacobsen, I. (1999). *The Unified Modeling Language User Guide*. Reading, MA: Addison-Wesley.

Brachman, R. J., & Levesque, H. J. (Eds.). (1985). *Readings in Knowledge Representation*. Los Altos, CA: Morgan Kaufman Publishers, Inc.

Branson, R. K., & Grow, G. (1987). Instructional Systems Development. In R. M. Gagné (Ed.), *Instructional Technology: Foundations*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Branson, R. K., Rayner, G. T., Cox, J. L., Furman, J. P., King, F. J., & Hannum, W. H. (1976). *Interservice Procedures for Instructional Systems Development* (Pamphlet 350-30). Fort Monroe, VA: U.S. Army, Training and Doctrine Command.

Carbonell, J. R. (1970). AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction. *IEEE Transactions on Man-Machine Systems*, *11*, 190-202.

Chi, M., Glaser, R., & Farr, M. (1988). *The Nature of Expertise*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Clancey, W. J. (1984). Extensions to Rules for Explanation and Tutoring. In B. G. Buchanan & E. H. Shortliffe (Eds.), *Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison-Wesley.

Clancey, W. J. (1985). *Acquiring, Representing, and Evaluating a Competence Model of Diagnostic Strategy* (HPP-84-2; STAN-C5-85-1067). Palo Alto, CA: Stanford University, Department of Computer Science.

Clancey, W. J. (1986). From GUIDON to NEOMYCIN and HERACLES in Twenty Short Lessons: ORN Final Report 1979 - 1985. *The AI Magazine*, *August*, 40-60.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive Apprenticeship: Teaching the Craft of Reading, Writing, and Mathematics. In L. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cook, D. A. (1997). Behavioral Analysis as a Basis for Instructional Design. In C. R. Dills & J. A. Romiszowski (Eds.), *Instructional Development Paradigms*. Englewood Cliffs, NJ: Educational Technology Publications.

Drake, L. D. (1997). *Design and Development of a Computer-Based Simulation Authoring System for Problem-Solving Instruction.* Unpublished Doctor of Philosophy, Utah State University, Logan, UT.

Edelson, D. C. (1998). Learning From Stories: An Architecture for Socratic Case-Based Reasoning. In R. C. Schank (Ed.), *Inside Multi-media Case-Based Instruction*. Mahwah, NJ: Lawrence Erlbaum Associates.

Fletcher, J. D. (1999). *Intelligent Tutoring Systems: Then and Now*. Hampton, VA: NASA Langley Research Center.

Gagné, R. M. (1965). The Analysis of Instructional Objectives for the Design of Instruction. In R. Glaser (Ed.), *Teaching Machines in Programed Learning, II: Data and Directions*.: Association for Educational Communications & Technology.

Gagné, R. M. (1968). Learning Hierarchies. *Educational Psychologist, 6*, 1-9.

Gagné, R. M. (1985). *The Conditions of Learning and Theory of Instruction* (4th ed.). Orlando, FL: Holt, Rinehart, and Winston, Inc.

Gibbons, A. S. (1977). *A Review of Content and Task Analysis Methodolgy* (2). San Diego, CA: Courseware, Inc.

Gibbons, A. S. (1998). Model-centered instruction, *Annual meeting of the american educational research association*. San Diego, CA.

Gibbons, A. S., & Fairweather, P. G. (1998). *Computer-Based Instruction: Design and Development*. Englewood Cliffs, NJ: Educational Technology Publications.

Gibbons, A. S., & Fairweather, P. G. (2000). Computer-Based Instruction. In S. Tobias & J. D. Fletcher (Eds.), *Training and Retraining: Handbook for Business, Industry, Government, and Military*. New York: Macmillan Reference USA.

Gibbons, A. S., Fairweather, P. G., Anderson, T. A., & Merrill, M. D. (1997). Simulation and Computer-Based Instruction: A Future View. In C. R. Dills & J. A.

Romiszowski (Eds.), *Instructional Development Paradigms*. Englewood Cliffs, NJ: Educational Technology Publications.

Gibbons, A. S., Lawless, K., Anderson, T. A., & Duffin, J. R. (in press). The Web and Model-Centered Instruction. In B. H. Khan (Ed.), *Web-Based Training*. Englewood Cliffs, NJ: Educational Technology Publications.

Gibbons, A. S., & Lester, S. M. (1996). *A Rexamination of the Media Selection Process: The Impact of New Instructional Forms and Approaches on the Selection of Instructional Media*. Idaho Falls, ID: Center for Performance Improvement, INEEL.

Gibbons, A. S., & Nelson, J. S. (1999). *Model-Centered Analysis Process (MCAP): A Pre-Design Analysis Methodology*. Idaho Falls, ID: Idaho National Engineering and Environmental Laboratory.

Gibbons, A. S., & O'Neal, A. F. (1989). New Work Surfaces for Instructional Design. *Journal of Interactive Instructional Devlopment, Spring*, 16-20.

Glaser, R. (Ed.). (1965). *Teaching Machines in Programed Learning, II: Data and Directions*.: Association for Educational Communications & Technology.

Guidelines for Evaluation of Nuclear Facility Training Programs (DOE-STD-1070-94)(1994). Washington, D.C.: U.S. Department of Energy.

Guilford, J. P., & Hoepfner, R. (1971). *The Analysis of Intelligence*. New York: McGraw-Hill.

Hall, E. P., Gott, S. P., & Pokorny, R. A. (1995). *Procedural Guide to Cognitive Task Analysis: The PARI Methodology* (AL/HR-TR-1995-0108). Brooks, AFB, TX: Armstrong Laboratory, Human Resource Directorate.

Horn, J. L. (1989). Cognitive Diversity: A Framework of Learning. In P. L. Ackerman & R. J. Sternberg & R. Glaser (Eds.), *Learning and Individual Differences: Advances in Theory and Research*. New York: W. H. Freeman.

Jarke, M. (1998). Requirements Tracing. *Communications of the ACM, 41*(2).

Jonassen, D. H., & Hannum, W. H. (1991). Analysis of Task Analysis Procedures. In G. J. Anglin (Ed.), *Instructional Technology: Past, Present, and Future* (2nd ed.). Englewood, CO: Libraries Unlimited.

Kieras, D. E. (1988). What Mental Models Should Be Taught: Choosing Instructional Content for Complex Engineered Systems. In J. Psotka & D. L. Massey & S. A. Mutter (Eds.), *Intelligent Tutoring Systems: Lessons Learned* (pp. 85-111). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kyllonan, P. C., & Shute, V. J. (1989). A Taxonomy of Learning Skills. In P. L. Ackerman & R. J. Sternberg & R. Glaser (Eds.), *Learning and Individual Differences: Advances in Theory and Research*. New York: W. H. Freeman.

Lange, P. C. (1967). *Programmed Instruction: The 66th Yearbook of the National Society for the Study of Education*, Chicago, IL.

Lesgold, A. M. (1993). An Object-based Situational Approach to Task Analysis. In M. Caillot (Ed.), *Learning Electricity and Electronics with Education Technology* (Vol. 15, pp. 291-301).

Lesgold, A. M. (1999). Intelligent Learning Environments for Technical Training: Lessons learned. In A. K. Noor (Ed.), *Workshop on Advanced Training Technologies and Learning Environments*. Hampton, VA: NASA Langley Research Center.

Loughner, P., & Moller, L. (1998). The Use of Task Analysis Procedures by Instructional Designers. *Performance Improvement Quarterly, 11*(3), 79-101.

Lumsdaine, A. A., & May, M. A. (1965). Mass Communication and Educational Media. *Annual Review of Psychology, 16*, 475-534.

Means, B., & Gott, S. P. (1988). Cognitive Task Analysis as a Basis for Tutor Development: Articulating Abstract Knowledge Representations. In J. Psotka & D. L. Massey & S. A. Mutter (Eds.), *Intelligent Tutoring Systems: Lessons Learned*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Melton, A. W. (Ed.). (1964). *Categories of Human Learning*. New York: Academic Press, Inc.

Merrill, M. D. (1999). Instructional Transaction Theory (ITT): Instructional Design Based on Knowledge Objects. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory*. Mahwah, NJ: Lawrence Erlbaum Associates.

Merrill, M. D., Li, Z., & Jones, M. K. (1991). Instructional Transaction Theory: An Introduction. *Educational Technology, 31*(6), 7-12.

Merrill, M. D., & Twitchel, D. G. (Eds.). (1993). *Instructional Design Theory*. Englewood Cliffs, NJ: Educational Technology Publications.

Merrill, P. F. (1987). Job and Task Analysis. In R. M. Gagné (Ed.), *Instructional Technology: Foundations*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Militello, L. G., & Hutton, R. J. B. (1998). Applied Cognitive Task Analysis (ACTA): A Practitioner's Toolkit for Understanding Cognitive Task Demands. *Ergonomics, 41*(11), 1618-1641.

Miller, R. B. (1963). Task Description and Analysis. In R. M. Gagné (Ed.), *Psychological Principles in Systems Development*. New York: Holt, Rinehart, and Winston, Inc.

Miller, R. B. (1971). *Development of a Taxonomy of Human Performance: Design of a System Task Vocabulary* (II). Washington, D.C.: American Institute for Research.

Montague, W. E. (1988). Promoting Cognitive Processing and Learning by Designing the Learning Environment. In D. H. Jonassen (Ed.), *Instructional Design for Microcomputer Courseware*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, Inc.

Psotka, J., Massey, D. L., & Mutter, S. A. (Eds.). (1988). *Intelligent Tutoring Systems: Lessons Learned*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Regian, J. W. (1999). Deployable Agents for Training, Aiding, and Guidance. In A. K. Noor (Ed.), *Workshop on Advanced Technologies and Learning Environments.* Hampton, VA: NASA Langley Research Center.

Resnick, L., Wang, M. C., & Kaplan, J. (1970). *Behvioral Analysis in Curriculum Design: A Hierarchically Sequenced Introductory Mathematics Curriculum*. Pittsburgh, PA: University of Pittsburgh Learning Research and Development Center.

Resnick, M. (1997). *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Cambridge, MA: MIT Press.

Resnick, M., & Silverman, B. (1997, 97/8/24). *Active Essays,* [Web page]. Available: http://el.www.media.mit.edu/groups/el/projects/circles/active-essay.html.

Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Phillips, J., Jackiw, N., &

Suthers, D. (1999). Developing Educational Software Components. *IEEE Computer*, *32*, 50-58.

Rothkopf, E. Z. (1996). Control of Mathemagenic Activities. In D. H. Jonassen (Ed.), *Handbook of Research for Educational Communications and Technology* (pp. 879-896). New York: Simon & Schuster Macmillan.

Schank, R. C. (1998). *Tell Me a Story: Narrative and Intelligence*. Evanston, IL: Northwestern University Press.

Schank, R. C., & Fano, A. (1992). *A Thematic Hierarchy for Indexing Stories*. Evanston, IL: The Institute for the Learning Sciences, Northwestern University.

Schank, R. C., Kass, A., & Riesbeck, C. K. (Eds.). (1994). *Inside Case-Based Explanation* (Vol. 3). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schraagen, J. M. C., Chipman, S. E., Shute, V. J., Annett, J., Strub, M., Sheppard, C., Ruisseau, J., & Graff, N. (1997). *State-of-the-art Review of Cognitive Task Analysis Techniques* (RSG.27): TNO Human Factors Research Institute Group: Information Processing.

Skinner, B. F. (1953). *Science and Human Behavior*. New York: Free Press.

Taylor, B., & Ellis, J. (1991). An Evaluation of Instructional Systems Development in the Navy. *Educational Technology Research and Development*, *39*(1), 93-103.

Wedman, J., & Tessmer, M. (1993). Instructional Designer's Decisions and Priorities: A Survey of Design Practices. *Performance Improvement Quarterly*, *6*(2), 43-57.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufman Publishers, Inc.

White, B. Y., & Frederiksen, J. R. (1990). Causal Model Progressions as Foundations for

Intelligent Learning Environments. *Artificial Intelligence, 42,* 99-157.

Winer, L. R., & Vasquez-Abad, J. (1995). The Present and Future of Instructional Design Practice. *Performance Improvement Quarterly, 8*(3), 55-67.

Wittrock, M. C. (1974). Learning as a Generative Process. *Educational Psychologist, 11*, 87-95.

Youngblut, C. (1994). *Government-Sponsored Research and Development Efforts in the Area of Intelligent Tutoring Systems* (IDA Paper P-3003). Alexandria, VA: Institute for Defense Analyses.