

AD-A244 062



2

**DESIGNING AN ADVANCED INSTRUCTIONAL
DESIGN ADVISOR: TRANSACTION SHELL
THEORY (VOLUME 6 OF 6)**

M. David Merrill

Department of Instructional Technology
Utah State University
Logan, UT 84322

54 b

J. Michael Spector

**HUMAN RESOURCES DIRECTORATE
TECHNICAL TRAINING RESEARCH DIVISION
Brooks Air Force Base, TX 78235-5000**

December 1991

Interim Technical Paper for Period August 1989 - February 1991

Approved for public release; distribution is unlimited.

91-19156



91 1227 024

**AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5000**

ARMSTRONG

LABORATORY

DTIC
DEC 30 1991

NOTICES

This technical paper is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

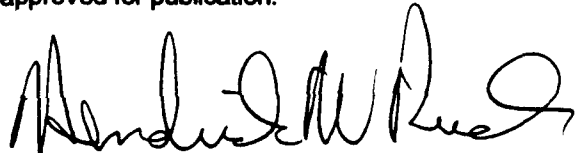
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.



J. MICHAEL SPECTOR
Project Scientist



HENDRICK W. RUCK, Technical Director
Technical Training Research Division



RODGER D. BALLENTINE, Colonel, USAF
Chief, Technical Training Research Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1991		3. REPORT TYPE AND DATES COVERED Interim - August 1989 - February 1991	
4. TITLE AND SUBTITLE Designing an Advanced Instructional Design Advisor: Transaction Shell Theory (Volume 6 of 6)				5. FUNDING NUMBERS C - F33615-88-C-0003 PE - 62205F PR - 1121 TA - 10 WU - 43	
6. AUTHOR(S) M. David Merrill J. Michael Spector					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Utah State University Department of Instructional Technology Logan, UT 84322				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES) Armstrong Laboratory Human Resources Directorate Technical Training Research Division Brooks Air Force Base, TX 78235-5000				10. SPONSORING/MONITORING AGENCY REPORT NUMBER AL-TP-1991-0017-Vol-6	
11. SUPPLEMENTARY NOTES These papers were submitted in conjunction with Task Order #0006 in WU 1121-10-43; they do not constitute the interim or final report.					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The Advanced Instructional Design Advisor (AIDA) is an R&D project being conducted by the Armstrong Laboratory Human Resources Directorate and is aimed at producing automated instructional design guidance for developers of computer-based instructional materials. The process of producing effective computer-based instructional materials is complex and time-consuming. Few experts exist to insure the effectiveness of the process. As a consequence, the Air Force is committed to providing its courseware developers with up-to-date guidance appropriate for the creation of computer-based instruction. The assistance should be provided in an integrated automated setting. This paper addresses the nature of the automated setting in which the assistance is to be provided.					
14. SUBJECT TERMS automated design advisor cognitive learning theory instructional design theory				15. NUMBER OF PAGES 72	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	
				20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. THE AIDA CONCEPTUAL BACKGROUND	3
Background	3
Concept of an AIDA System	4
Theoretical Foundations for an AIDA	11
III. A THEORETICAL BASIS FOR TRANSACTIONS	28
Introduction	28
Metatheory -- A Very Abbreviated Theory of Theories.	28
Theoretical Interpretation	29
Prologue	39
IV. ELABORATION OF THE AIDA CONCEPT	40
Introduction	40
Information Component	42
Content Component	43
AIDA Executive	46
Strategies Component	47
Delivery Component	54
Evaluation Component	55
V. CONCLUSION	57
REFERENCES	58

LIST OF FIGURES

Figure	Page
1 General AIDA Functional Architecture	1
2 Components of ID EXPERT v3.0	5

LIST OF TABLES

Table	Page
1 Comparison of Instructional Objects	29

PREFACE

The work reported herein was done for the Advanced Instructional Design Advisor project at the Air Force Armstrong Laboratory (AL/HRT). The substance of this research was done under contract to Mei Associates, Inc., the primary contractor on the Advanced Instructional Design Advisor (Contract No. F33615-88-C-0003).

This work was done as part of the first phase effort on the Advanced Instructional Design Advisor. The initial phase of this project established the conceptual framework and functional specifications for the Advanced Instructional Design Advisor, an automated and intelligent collection of tools to assist subject matter experts who have no special training in instructional technology in the design and development of effective computer-based instructional materials.

Mei Associates' final report for the initial phase will be published as an Armstrong Laboratory Technical Paper. In addition, Mei Associates received 14 papers from the seven consultants working on this phase of the project. These 14 papers have been grouped into six sets and edited by AL/HRT personnel. They are published as Volumes 1 - 6 of Designing an Advanced Instructional Design Advisor:

Volume 1: Cognitive Science Foundations
(AL-TP-1991-0007)

Volume 2: Principles of Instructional Design
(AL-TP-1991-0017)

Volume 3: Possibilities for Automation
(AL-TP-1991-0008)

Volume 4: Incorporating Visual Materials and Other
Research Issues (AL-TP-1991-0017-Vol-4)

Volume 5: Conceptual Frameworks (AL-TP-1991-0017-Vol-5)

Volume 6: Transaction Shell Theory
(AL-TP-1991-0017-Vol-6)

This is Volume 6 in the series. Dr. J. Michael Spector wrote Sections I and V. Dr. M. David Merrill wrote Sections II, III, and IV.

SUMMARY

The Advanced Instructional Design Advisor is an R & D project being conducted by the Air Force Armstrong Laboratory (Human Resources Directorate) in response to an Air Training Command (ATC) Manpower, Personnel, and Training Need calling for improved guidelines for authoring computer-based instruction (CBI) (MPTN 89-14T).

Aggravating the expensive and time-consuming process of CBI development is the lack of Air Force personnel who are well-trained in the areas of instructional technology and educational psychology. More often than not, a subject matter expert with little knowledge of CBI is given the task of designing and developing a computer-based course. Instructional strategies that work in a classroom are often inappropriate in a computer-based setting (e.g., leading questions may work well in a classroom but are difficult to handle in a computer setting). Likewise, the computer offers the capability to present instruction in ways that are not possible in the classroom (e.g., computer simulations can be used to enhance CBI).

The Advanced Instructional Design Advisor is a project aimed at providing subject matter experts who have no background in computer-based instructional systems with automated and intelligent assistance in the design and development of CBI. The goal is to reduce CBI development time while insuring that the instructional materials are effective.

I. INTRODUCTION (Spector)

The Advanced Instructional Design Advisor is an R & D project aimed at providing automated and intelligent assistance to inexperienced instructional designers who have the task of designing and developing computer-based instruction (CBI). The particular problem being addressed by this line of research is the need for more cost efficient methodologies for the design and development of CBI. Current methods for developing CBI are expensive, time-consuming, and often result in ineffective instruction due to the general lack of expertise in computer-based instructional systems (Spector, 1990).

The Advanced Instructional Design Advisor project is divided into four phases:

Phase 1: Conceptualization & Functional Specifications

Phase 2: Conceptual Refinement & System Specifications

Phase 3: Prototype, Field Test, & Refinement

Phase 4: Technology Demonstration & System Validation

The first two phases have been performed under Task Order Contracts. The third phase is being accomplished via a Broad Agency Announcement (BAA). The fourth phase will be funded by a fully specified contract. The work reported herein concerns the first phase.

One primary result of the initial phase effort was agreement on the functional characteristics of an Advanced Instructional Design Advisor. The functional concept of AIDA is depicted in Figure 1 below:

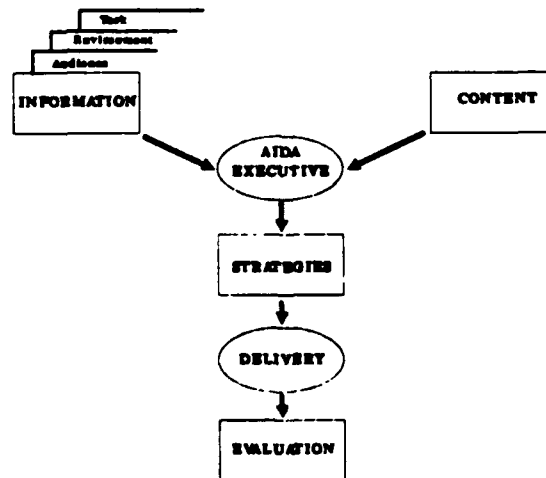


Figure 1. General AIDA Functional Architecture

AIDA Form	
AIDS-000001	
BIO-000	
Unit-000	
JAN-000	
By: [Signature]	
Date: [Date]	
[Blank]	
[Blank]	
A-1	

The next three sections of this paper reflect the contributions of one of the seven academic consultants to the project: M. David Merrill, Professor of Instructional Technology at Utah State University. These sections represent papers presented by Professor Merrill at AIDA Technical Interchange Meetings. As a consequence, the concepts undergo a subtle evolution and growth that reflects the contributions of the entire AIDA development team (Drs. Robert Gagné, Henry Halff, Martha Polson, Harry O'Neil, Charles Reigeluth, and Robert Tennyson, plus numerous DoD advisors).

In Section II, Merrill provides a historical view of the development of ID Expert, versions 1 through 3. He elaborates the architecture of ID Expert v3.0 and suggests that it is appropriate to consider it as the beginning point for an AIDA.

In Section III, Merrill presents a theoretical discussion of transaction theory, which underlies ID Expert v3.0 and his view of AIDA. He explains in detail the inadequacies of first generation instructional design theories, and he proposes a second generation theory that takes into account recent developments in cognitive science and instructional technology.

In Section IV, Merrill provides an elaboration of the variables, parameters, and attributes associated with the AIDA architecture represented in Figure 1.

These three sections are generally representative of Merrill's more recent work in the area of automated instructional design. One way to summarize Merrill's position is this: Once the instructional design has been completed, the process of producing the instruction and delivering it to students can be automated.

Merrill's proposed AIDA, as refined by the AIDA development team, largely represents the current direction of the AIDA project. Other types of instructional design automation are possible and were considered by the AIDA development team. For example, Gagné proposed a simpler system that automated the process of providing instructional design guidance to the SME/designer (see Volume 5 in this series). He is pursuing this concept (GAIDA -- Guided Approach to Instructional Design Advising) independently as part of a National Research Council Senior Fellowship with the Laboratory. Tennyson proposed a more complex system which, in essence, was an intelligent tutoring system for the domain of designing instruction for computer-based settings (see Volume 5 in this series).

The final section of this paper contains a summary by the AIDA project manager, Dan Muraida. A second series is planned for the papers delivered in the second phase of the project.

II. THE AIDA CONCEPTUAL BACKGROUND (Merrill)

[NOTE: In the next three sections, 'we' refers to the Utah State University Instructional Technology Development Team which includes M. David Merrill, Mark K. Jones, and Zhongmin Li.]

The purpose of this section is to provide an initial concept of AIDA: the functions it should perform, including knowledge acquisition and strategy analysis. An attempt is also made to identify the principles of knowledge theory, learning theory, and instructional theory that underlie these functions.

Background

For the past two and one-half years in cooperation with Human Technology Inc., supported in part from funds from the Army Research Institute and the Office of Personnel Management, the ID EXPERT project team (M. David Merrill, Zhongmin Li and Mark K. Jones) have explored the construction of an Instructional Design Expert System. We have built two prototype systems, ID EXPERT v1.0 and ID EXPERT v2.0 (Merrill 1987b, 1987c; Merrill & Li, 1988, 1989a, 1989b).

ID EXPERT v1.0 was implemented on a VAX computer using the expert system shell S.1. This prototype was primarily rule based with a linear interface. It did, however, demonstrate the feasibility of an instructional design expert. This v1.0 prototype can guide a limited content analysis, make reasonable recommendations for course organization and make reasonable recommendations about possible transactions within this course organization for a limited domain of subject matter. This prototype explored various rule structures including the accumulation of evidence using certainty factors for decisions involving many attributes.

ID EXPERT v2.0 transported the expertise to a desktop platform using Macintosh SE computers and the expert system shell NEXPERT with a HYPERCARD interface. This prototype shifted from a rule-based system to a hybrid system which is primarily frame-based. The object-oriented programming characteristics of HYPERCARD, together with the object-oriented implementation of NEXPERT were exploited to represent instructional objects. As explained in the "Knowledge Representation" section of this paper, a frame representation is more appropriate for a design task. Rules are used to select between alternative frames. This system expanded content analysis to a preliminary version of a knowledge acquisition system. We began to explore the concept of knowledge acquisition explained in the "Conceptual Approach" section of this paper. This prototype and related work explored transactions and several sample transaction instances were

constructed. Furthermore, the interface was improved to be more graphic and interactive allowing a nonlinear navigation through the instructional design process.

Version 2.0 is continuing to be improved. A more consistent Human Computer Interface (HCI) has been designed but not yet implemented. The requirements for a more complete Knowledge Analysis and Acquisition System (KAAS) have been specified. Preliminary designs for implementation of this Knowledge Analysis and Acquisition System have been completed. Support for this work has come from the National Security Agency in cooperation with the US Air Force Academy and from IBM Corporation. We are continuing to seek additional funding to support this effort. Obviously the AIDA concept is very similar to the work we are doing and we seek ways for cooperative effort with HRL.

In the following section we will provide a preliminary description of the system on which we are working. Hopefully these ideas will have benefit for the AIDA project and for discussion by the MEI team of consultants.

Concept of an AIDA System

ID EXPERT v3.0 is based on the premise that existing ISD theory and methodology, first generation instructional design (ID-1), lacks prescriptions that are necessary for the design of instruction for interactive technologies (computer-based instruction, interactive video disc, intelligent tutoring systems, etc.). An adequate intelligent instructional design system will require a second generation of ID theory (ID-2) which focuses attention on mental models, integrated learning events (transactions), experiential environments, and meaningful student interaction. In the last part of this chapter we have identified some of the assumptions (principles) underlying ID EXPERT v3.0 and the second generation instructional design theory on which it is based. ID-2 is more complex than ID-1. Its implementation will require the use of intelligent tools of which ID EXPERT v3.0 (or AIDA) is an example. These tools involve expert systems and other forms of artificial intelligence. The proposed system consists of a hybrid approach involving both frame-based and rule-based expertise.

Mini-Experts

In the early days of artificial intelligence research, efforts were directed towards developing a general problem solver, capable of dealing with any situation. The first breakthroughs in artificial reasoning came, however, when the focus shifted to the design of systems limited to a specific, highly constrained domain. The typical expert system today contains a large rule base, and an inference engine that applies

these rules to available data to reach decisions or to make recommendations. These rule bases, however, tend to be monolithic, and directed towards a single decision or set of decisions. The instructional design process, on the other hand, is not one decision but by many different decisions. For this reason, we choose to represent pedagogic expertise in a set of mini-experts, each of which functions independently at different parts of the process, and each of which is responsible for a relatively narrow decision. ID-2 will prescribe the function of these mini-experts and provide a means by which the various individual decisions can be coordinated and combined to make the larger decisions involved at various steps in the process of design.

Components of ID EXPERT v3.0 (a possible AIDA)

ID EXPERT v3.0 consists of four principal subsystems: (1) a Knowledge Acquisition and Analysis System (KAAS), (2) a Strategy Analysis System (SAS), (3) a set of transaction subsystems including a Transaction Generation System (TGS), a Transaction Configuration System (TCS) and a Transaction Library (TLIB), and (4) an Intelligent Advisor based Instructional Delivery System (IADV). These components are depicted in the following figure:

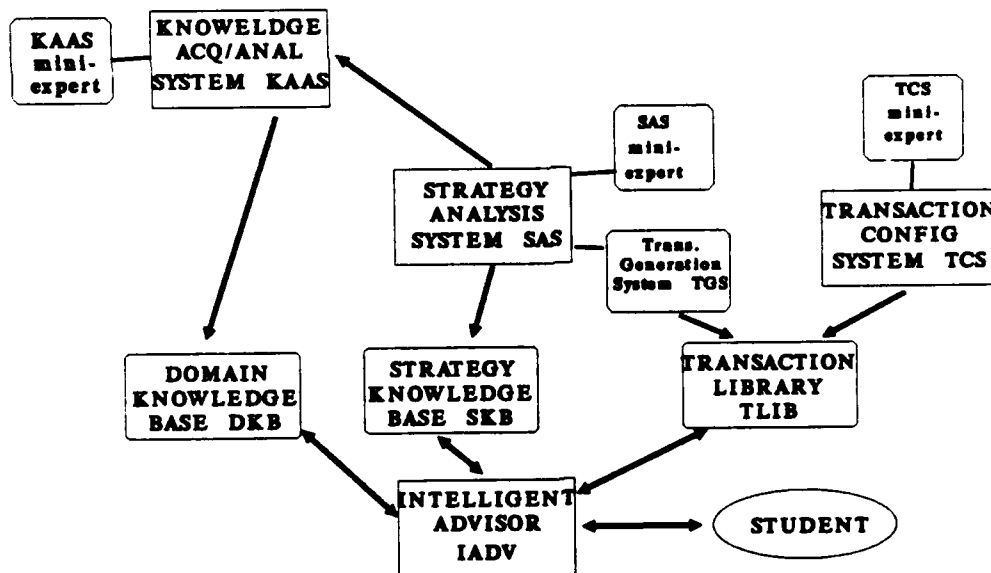


Figure 2. Components of ID EXPERT v3.0

KAAS replaces the ISD functions of content analysis and extends ISD by providing knowledge acquisition based on mental models. SAS replaces the ISD functions of strategy analysis and extends ISD by integrating KAAS with instructional delivery via course organization and configuration using transaction classes which correspond to mental models. TGS, TCS and TLIB replace traditional authoring systems and focus on integrated student interaction with transaction classes and individual transactions. IADV provides an intelligent coach-based interactive delivery system. Each of these subsystems are described in more detail in the following paragraphs.

Knowledge Acquisition and Analysis System (KAAS)

The knowledge analysis and acquisition system guides the user in providing information about the subject matter to be taught. This system consists of frames for different content structures. "Frames" here refer to entities defined by the artificial intelligence community consisting of slots and required legal values for these slots. We are not referring to instructional displays or programmed instruction frames.

A given content structure frame knows the necessary knowledge components required for its instantiation. This knowledge includes the components of the content structure frame, the level of abstraction (instance, class or superclass) associated with the content structure frame, and the rules for inheritance from one abstraction level to another.

In addition KAAS knows the possible links between various frames and how to propagate knowledge from one frame to a linked frame. The knowledge base (rules) underlying KAAS is used to prompt the designer/user to supply the necessary values for various content structure frame slots. The designer/user is led to identify the frames, frame abstraction level, frame components, and frame links necessary to describe the subject matter content to be taught. This subject matter information comprises the domain knowledge base which is built by KAAS.

Prescriptive Mini-experts for KAAS

KAAS will be accompanied by a series of mini-experts which will guide the knowledge acquisition process. These mini-experts will guide the following decisions: frame type (entity, activity, process), frame subtype (type of entity, type of activity, type of process), required frame components, appropriate level of abstraction for a given frame, required and/or desirable links for a given frame. Any time the user is required to make a technical decision or select an appropriate parameter value, ID EXPERT will provide a mini-expert to assist with this decision or selection.

For all prescriptions based on rules contained in the mini-experts, the user/designer will be able to ask for an explanation of the recommendation. The explanation will consist of multiple levels. At the first level the reasoning will be explained showing the attributes and values which were involved in the reasoning. At a second level the prescriptions (propositions) which comprise the built-in knowledge of the system will be presented and their application explained. At a third level references to relevant theory and research will be provided which will enable the user/designer to study the sources on which the system is built.

Strategy Analysis System (SAS)

The strategy analysis system provides a strategy link between knowledge acquisition and transactions. SAS queries the user/designer to obtain specific information about subject matter goals, learner characteristics and environmental constraints. Using its build-in strategy rules and the information provided by the user/designer, SAS also knows how to recommend course organization and transaction strategy. Course organization is a sequence of transaction frame sets, each instantiated with appropriate knowledge and configured for the specified goals, environment and learner audience. Transaction strategy is the particular order of individual interactions within a given transaction frame set.

SAS also knows how to provide the necessary course and transaction management information. Management information provides the control of sequence at the various decision points in the course (between transaction frame sets) or in the transaction frame set (between individual interactions). This control can vary from learner control via a menu, to system control via some criterion of performance. Note that this control is much more than simple branching between displays as is typical of much CBI or programmed instruction. The criteria used here may consist of a history of responses over a whole transaction frame set or several such sets as well as a single response to a single display. Furthermore, the decision is between complete interaction sequences rather than between individual displays. All strategy decisions are stored in a strategy data base attached to a course.

Interrelations of SAS and KAAS

SAS uses its built-in strategy knowledge (rules) and the information provided by the user/designer to provide both filters and prescriptions for KAAS. KAAS knows about kinds of knowledge structure frames and how to acquire the required information from the user/designer, but it does not know when to stop or when to get more knowledge. A filter supplied by SAS indicates that certain knowledge is not necessary for the specified

instructional goals or learners and enables the system to suggest that certain subject matter knowledge may not be necessary. A prescription supplied by SAS indicates that for the goals and students under consideration certain knowledge structure frames are required, certain links between these elaborated frames are required, and specific knowledge must be propagated from one knowledge structure frame to another in order to have the knowledge required by the specified learning goal and learner population.

Prescriptive Mini-experts for SAS

SAS will also be accompanied by a series of mini-experts which will guide the strategy analysis process. These mini-experts will guide the following information gathering decisions: (1) parameter values for various student attributes such as motivation, familiarity, aptitude, role to be played, and other attributes; (2) parameter values for various environmental decisions such as appropriate delivery system(s), instructional setting, time limitations, and other environmental characteristics; and (3) parameter values for goals such as type of goal and subgoal, student performance speed and/or accuracy, level of performance, and other goal characteristics.

Other mini-experts will guide the designer/user in selecting from course organization and management options recommended and in selecting from various transaction frame sets or various equivalent transactions within a transaction frame set.

As indicated above, all mini-experts will be accompanied by a tri-level explanation system.

Transaction Generation System (TGS)

Associated with each type and/or subtype of knowledge structure frame is a transaction class (or several transaction classes). A transaction class includes all of the possible (or more practically all of the implemented) transaction frame sets which are appropriate for teaching the knowledge contained in a particular type of knowledge structure frame. A transaction frame set is the set and sequence of interactions thought most appropriate to teach the content of a particular implementation of a knowledge structure frame. A transaction frame set can be composed of many different individual transactions or interactions. The TGS selects and recommends a particular transaction frame set and a particular transaction strategy (set and sequence of individual transactions) for this transaction frame set. This recommendation is based on characteristics of the goal, characteristics of the student population and characteristics of the environment in which the instruction will be delivered.

The TGS implements a dynamic object-oriented design of transactions, under control of the SAS. A class "transaction" is defined, with a subclass structure beneath it corresponding to the different types of possible transactions that can be generated by the system that are appropriate for this class. A given transaction is instantiated by the TGS from this class hierarchy, and the capabilities of the transaction instance are inherited through the class structure. A set of such transactions comprise a transaction frame set. The transaction instances comprising the transaction frame set are those interactions thought to be appropriate for promoting the acquisition of a particular elaborated frame (corresponding to a mental model) from the knowledge structure created by KAAS in cooperation with SAS.

Transaction Configuration System (TCS)

The individual transactions of a transaction frame set are really "interaction shells" which can be used with different instances of the subject matter best taught by the transaction. In addition each individual transaction has several specific parameters which best configure the interaction for a particular student or particular content instance. These parameters can be adjusted during the design process providing a default setting for the interaction or they can be adjusted dynamically while the instruction is in progress by the intelligent advisor system. This customizes the instruction for a particular student while the instruction is in progress.

The transaction configuration system provides a means to customize the default values for individual transactions for inclusion in a course. It will work with a transaction library, which consists of a set of transaction instances (shells) each of which knows how to teach some part of a particular type of knowledge structure frame. Combined with other transaction instances a transaction frame set includes all the instruction necessary to promote the acquisition of a given knowledge structure frame. A given transaction instance knows what knowledge it must have in order to teach. It will be able to query the content knowledge base to find the required knowledge and thus be able to instantiate its knowledge slots with this information. If the content knowledge base does not contain the necessary knowledge the TCS can direct the user/designer to supply the required information via KAAS. A transaction instance can be adapted to a variety of environmental constraints and student characteristics by means of a set of customization parameters. It can query the strategy data base created by the strategy analysis system or it can query the user/designer directly to obtain specific values for each of these parameters. During the instruction the transaction instance can query the intelligent advisor for student characteristics necessary to customize the interaction on the fly.

Prescriptive Mini-experts for TCS

TCS will be accompanied by a series of mini-experts which will assist the user designer in making configuration decisions. These will include mini-experts for selecting values for screen display parameters, timing parameters, type and frequency of feedback parameters and the many other variables of the individual interactions.

Transaction Library (TLIB)

Previously generated individual transactions will reside in a transaction library. The transaction generation system selects appropriate transaction instances from this library for inclusion in a given transaction frame set. In addition when the transaction generation system generates a transaction instance it is placed in the transaction library. Transactions can also be added to the transaction library from sources external to ID EXPERT for potential inclusion in transaction frame sets. During the instruction the intelligent advisor has access to the transaction library. If a given student needs a particular type of interaction, which was not originally included in the transaction frame sets of the course as it was designed, the advisor can select this transaction, configure it, and include it dynamically into the instruction for a particular student.

Intelligent Advisor (IADV) -- Transaction Customization

An intelligent advisor will be developed to customize instructional delivery. This version of the advisor will take some of the rules used by the strategy analysis system (SAS) and incorporate them for on-line use. SAS will prescribe a default path through the course organization. A default path is that sequence through the material, based on the information available before the commencement of the instruction, thought to be best for a given student. As the student progresses via this default path, performance data is accumulated. When this data indicates that the default path is not the optimal, the advisor will alter the sequence of transactions to more adequately adapt the instruction to the student.

The IADV employs a "weak" student model. It is not intended to be capable of either modeling a student's understanding of the domain in a way that can be executed, or in diagnosing misconceptions based on the model. The primary reason for not attempting this capability, in addition to the inherent difficulty, is the desire to have a domain-independent advisor.

Intelligent Advisor (IADV) -- Course Organization Customization

The level 1 advisor enables the student to deviate from the default path as justified by performance data gathered on-line.

An enhanced intelligent advisor would be more closely linked to the TGS and the content knowledge base so that in addition to changing sequence from a default path the advisor can enhance the instruction by selecting transactions not previously recommended by the course organization of SAS. This would essentially mean that the enhanced IADV could design a course organization and transaction sequence strategies on the fly. These new transactions would be instantiated, supplied with domain data from the content knowledge base, and their parameters configured on-the-fly as the instruction progressed. This enhanced intelligent advisor would essentially be the strategy analysis system (SAS) in real time.

Theoretical Foundations for an AIDA

First Generation Instructional Design (ID-1)

The use of contemporary instructional design methodologies does result in instruction that is more effective than that based only on folklore and trial-and-error. However, these methods have not provided the hoped for increase in instructional effectiveness that would enable learners to more adequately and efficiently grasp and apply the content presented. Most are based on the psychology of the 50's and 60's; they are analytical, not synthetic; they are component rather than model or schema oriented; and their application requires considerable effort. Because the theories upon which these methods are based predates the development of highly interactive, technology-based delivery systems, little guidance is provided for developing instruction for these systems.

The most widely applied instructional design theory is based largely on the work of Robert M. Gagné and his associates at Florida State University. This work is often equated with the term Instructional Systems Development (ISD). It assumes a cumulative organization of learning events based on prerequisite relationships among learned behaviors. Gagné's principal assumption is that there are different kinds of learned outcomes, and that different internal and external conditions are necessary to promote each type. Gagné's original work (Gagné, 1965) was based on the experimental learning psychology of the time. This included concepts of paired associate learning, serial learning, operant conditioning, concept learning, and gestalt problem solving. Recent versions (Gagné, 1985) have incorporated some ideas from cognitive psychology, but the essential characteristics of the original work have remained.

Our own work, Component Display Theory, (See Merrill 1983, 1987a, 1988) built directly upon Gagné's principal assumption. We extended the outcome classification system by separating content type from performance level. We also added a more

detailed taxonomy of presentation types and clarified the prescriptions of the Gagné position. Nevertheless, Component Display Theory has the same roots as the Gagné position.

Other contemporary instructional design theories (see Reigeluth, 1983, 1987) are consistent with the Conditions of Learning and Component Display Theory. Gagné extends cumulative prerequisite analysis by including Information Processing Analysis as suggested by Paul Merrill (Gagné, 1985). The recommendations for Structural Analysis by Scandura (Scandura, 1983; Stevens and Scandura, 1987) and Algorithm/Heuristic Analysis by Landa (1983, 1987) are similar to Information Processing Analysis. Markle (1983), Gropper (1983, 1987), Engelmann & Carnine (1982) and Collins (Collins & Stevens, 1983; Collins, 1987) provide a set of recommendations for teaching concepts and rules that are similar to the recommendations of The Conditions of Learning and Component Display Theory. Most of these theories were developed independently of one another, yet produce similar recommendations. This provides some rough confirmation of the validity of the recommendations.

In this paper we refer to this body of theory and methodology as First Generation Instructional Design (ID-1). These First Generation ID Theories were preceded by a series of transitional theories including "Operant Conditioning" and "Programmed Instruction" based on Skinner (1953, 1957), the "Meaningful Verbal Learning Theory" of Ausubel (1963), and the instructional theories of Bruner (1966). While there is a remarkable similarity in their prescriptions, they share several limitations:

- content analysis focuses on components, not integrated wholes;

- limited prescriptions for knowledge acquisition;

- superficial prescriptions for course organization strategies;

- the theories are closed systems, asserting principles based on a subset of available knowledge, but not able to accommodate new knowledge as it becomes available;

- each phase of instructional development is performed independently of other phases, as the theories provide no means for integration or for sharing data;

- the resulting instruction teaches components but not integrated knowledge and skills;

- the resulting instruction is often passive rather than interactive;

and, these theories are very inefficient to use because an instructional designer must build every presentation from fundamental components.

Limitations of ID-1

Limitation 1. Content analysis does not use integrated wholes which are essential for understanding complex and dynamic phenomena. First generation instructional design methods attempt to identify the components of subject matter. These constituent components are then used to prescribe course organization and sequence. The elements of this analysis are individual content components such as facts, concepts, principles (rules) or procedures. The resulting instruction may be effective in teaching these pieces of the content, but is often not effective in helping students to integrate these components into meaningful wholes. Hence, students are able to pass exams but cannot apply the knowledge in a wider context. The sheer amount of knowledge which must be learned continues to accelerate. New scientific knowledge, especially, is often complex and dynamic. It is difficult to understand the complex interrelationships of knowledge with only isolated concepts and principles. An integrated understanding is essential. Cognitive psychology, in postulating the notion of schema or frame, suggests that cognitive structure consists of mental models. Learning results in the construction and elaboration of these models which serve to organize the knowledge, and to facilitate recall and further learning. No ID-1 content analysis procedure takes this notion of mental models into account.

Limitation 2. Limited prescriptions for knowledge acquisition. While ID-1 methods prescribe content structure as a result of the content analysis, none prescribe the subject matter components necessary to build a complete knowledge base for this structure. Hence, the resulting structures are little more than content outlines for which the designer must still gather considerable additional material in order to build the course.

The content structure resulting from content analysis is rarely used directly in the course materials. The form of representation, usually some diagram, is not in a form that can be used by the presentation. In fact, current design methodology often requires three different and separate specifications of the content: first, as a set of task descriptions or objectives; second, as a story board or script; and third, a program written in some computer or authoring language. In addition to being time-consuming, this separation of content analysis from course development decreases the correspondence between these two activities, resulting in course content that is not represented in the content structure or content structure elements that are not contained in the course materials.

Limitation 3. Limited prescriptions for course organization. For most ID-1 methods there is a gap between content analysis and course organization strategies. The prescription for course organization strategies is either not present or superficial. Prescriptions range from a one-to-one correspondence between content structure elements and instructional modules, to the bottom-up sequences suggested by Gagné's hierarchies. None of these ID-1 methods adequately accounts for different levels of instructional outcomes, such as familiarity versus basic instruction versus remediation. None of the ID-1 methods considers the highly interactive nature of the new technologies and how to prescribe highly interactive sequences.

Limitation 4. Existing theories are essentially closed systems. There is no means of incorporating fine-grained expertise about teaching and learning, gained from research, and applying this in the design process. While there remains much to understand about how people learn, we in fact know a great deal already. The designer of instruction must, however, apply this knowledge separately from the application of ID-1 theory, as no hooks are built into the theory to incorporate and apply new and better knowledge as it is discovered.

Limitation 5. ID-1 fails to integrate the phases of instructional development. Methodology based on ID-1 defines five phases of instructional development: analysis, design, development, implementation, and evaluation. While the outcomes of each phase are inputs to the next, and the development cycle is iterative, that is the extent of the integration of the phases. Separate tools are used, and separate knowledge representations are maintained in each phase. Theory provides no prescriptions for how changes made in one phase should lead directly to changes in another. For example, in the analysis phase, information about the content to be taught is gathered, and represented in terms of the tasks that are performed by someone skilled in the subject matter to be taught. In the design phase, learning objectives are developed for each task. While the task analysis is preliminary to the objectives development, theory does not prescribe how the task analysis should be used. Guidance is available to the designer on the form to write an objective, but its actual selection and content is a matter of judgment and experience. At the next phase, development, learning activities are designed for each objective. Again, guidance is limited to what should go into an activity; there is no prescription for selecting activities. Moreover, at this point there is no direct connection whatsoever between the task analysis and the learning activities, and no possibility that information could flow directly from the one to the other.

Limitation 6. ID-1 teaches pieces but not integrated wholes. Each of these ID-1 methods attempts to prescribe the

characteristics of the stimulus presentation to the student. These presentation components consist of elements such as definitions, examples, non-examples, practice problems, attention-focusing help, and prerequisite information. In every case the instructional designer must compose an instructional strategy from such elements to make a complete whole. Often these strategies take on a disjointed character in which one content element is taught after another, but little is done to integrate a series of elements into a whole. Elaboration theory (Reigeluth, 1983, 1987) is an exception to other first generation theories in that it does attempt to provide some integration.

Limitation 7. Instruction is often passive rather than interactive. Most of the ID-1 theories were formulated before interactive media (computer based instruction, interactive video, intelligent tutoring systems) were readily available. As a consequence most of these models concentrate on the stimulus elements of the presentation rather than on input elements. Instruction based on ID-1 is frequently passive rather than interactive, requiring little mental effort on the part of the student. ID-1 theories are display oriented (our own work is called Component Display Theory) rather than transaction or interaction oriented. Collins' (1983,1987) inquiry-based prescriptions is the only theory in the Reigeluth collection that is concerned with dynamic on-line adaptation of the instruction based on student interaction with the materials. They prescribe examples and non-examples but have little to say about the use of experiential interactions, simulated environments, or controllable worlds (see Merrill, 1988).

There is evidence that learning is directly related to the level of mental effort put forth by the student. This mental effort must bear a direct relationship to the concepts and principles being taught. When the instruction is passive, learners are not forced to examine their cognitive structure and the resulting learning is poorly retained, does not relate well to previously learned materials, and is not easily transferred to new situations. Furthermore, much new scientific knowledge is dynamic in character and cannot be understood without a more active representation and student involvement.

Limitation 8. Every presentation must be constructed from small components. With ID-1 methods the designer is forced to compose every instructional strategy from basic display elements, e.g., definitions, rules, examples, and helps. This means that for each lesson the designer must analyze and select every display element for presentation to the student. If one were to consider a larger content element, a mental model, then it is conceivable that there is a corresponding instructional transaction for promoting the acquisition of this mental model. Composing instruction from larger transaction units would mean considerable savings in development time and resources. By

analogy first generation instructional design is a little like limiting a chemist to the basic elements. The chemist can make anything but to get water you start with hydrogen and oxygen and make the compound first. We need some instructional compounds that can be used as wholes. However, none of the ID-1 methods identify such transaction wholes.

Limitation 9. Current ID is labor intensive. Current instructional design and development practices are extremely labor intensive. Even though the hardware is affordable, the courseware frequently is not. A development/delivery ratio of more than 200:1 is too high. The current ratio for designing and developing instruction for the new interactive technologies exceeds 200 hours of design/development for each 1 hour of delivered instruction (Lippert, 1989). Some estimates suggest ratios exceeding 500:1 just for programming.

The impact of computerization on other fields has been to increase productivity by reducing labor costs, or allowing greater production from the same labor. Personal computers probably owe their success to the electronic spreadsheet. Every financial planner could immediately see the efficiency of using an electronic spreadsheet. Tasks that at one time might require days or weeks could now be accomplished in minutes or hours.

In education and training the ratio is just the opposite. Educational experiences which can be planned and delivered in a few hours using conventional methods and technologies require days or weeks with the computer. It is often argued that the quality of the instruction justifies the increased effort. However, when data is gathered it often shows only a marginal advantage for the computer. This data rarely justifies the extra effort. Until now, computer-based instruction has only been cost effective when many students are taught by the same program over a considerable period of time, and the cost is justified by reducing personnel costs.

Proposed Solution - Second Generation Instructional Design ID-2

If interactive instructional technologies are to provide a significant part of the increasing amount of education and training demanded by society, then there is a critical need for significantly improved methodology and tools to guide the design and development of high quality interactive technology-based instructional materials. There is a need for second generation instructional design (ID-2).

ID-2 would build on the foundation of ID-1, but would address the shortcomings noted above. Specifically, ID-2 will:

be capable of analyzing, representing, and guiding instruction to teach integrated sets of knowledge and

skills;

be capable of producing pedagogic prescriptions for the selection of instructional strategies and the selection and sequencing of instructional transactions;

be an open system, able to incorporate new knowledge about teaching and learning and to apply these in the design process; and

integrate the phases of instructional development.

ID-2 will comprise the following components:

a theoretical base that organizes knowledge about instructional design and defines a methodology for performing instructional design;

a means of representing domain knowledge for the purposes of making instructional decisions;

a collection of mini-experts, each contributing a small knowledge base relevant to a particular instructional design decision or a set of such decisions;

a library of instructional transactions for the delivery of instruction, and the capacity to add new or existing transactions to the library; and

an on-line intelligent advisor program that dynamically customizes the instruction during delivery, based on a mixed-initiative dialog with the student.

Analyzing and Representing Instruction for Integrated Goals

Our orientation is cognitive rather than behavioral. We start from the basic assumption that learning results in the organizing of memory into structures, which we may term mental models. To this we adopt two propositions about the learning process:

organization during learning aids in later retrieval of information; and

elaborations generated at the time of learning new information can facilitate retrieval.

Organization refers to the structuring of knowledge, while elaboration refers to the explicit specification of relations among knowledge units.

From ID-1 we retain Gagné's fundamental assumption:

there are different learning outcomes and different conditions are required to promote each of these different outcomes (Gagné 1965, 1985).

We propose to extend these fundamental ideas as follows:

a given learned performance results from a given organized and elaborated cognitive structure, which we will call a mental model. Different learning outcomes require different types of mental models;

the construction of a mental model by a learner is facilitated by instruction that explicitly organizes and elaborates the knowledge being taught, during the instruction; and

there are different organizations and elaborations of knowledge required to promote different learning outcomes.

However, we make no claims about how cognitive structure is organized and elaborated, as this is not well understood. We stand on the weaker, and more defensible assumption, that we can analyze the organization and elaborations of knowledge outside the mind, and presume that there is some correspondence between these and the representations in the mind.

Addressing the limitations of ID-1 in regards to the teaching of integrated wholes, we propose that ID-2 should be capable of teaching the organized and elaborated knowledge needed to facilitate the development of mental models. A necessary precondition to the design of such instruction is the development of detailed prescriptions for a knowledge acquisition process to identify all of the information necessary for a student to build a mental model. The outcome of this process would be a representation of the knowledge to be taught in terms of its structure and its elaborations.

Classes of Knowledge Representations

The means chosen to represent knowledge about a domain depend upon the use to which that knowledge will be put. We distinguish for the purposes of this analysis three classes of knowledge representations (KR).

KR_r is a class of representation for the purpose of retrieving the knowledge in various formats. A representation of this class would be most appropriate for database applications, and would emphasize descriptors, keys, and relations.

KR_e is the class most often used in artificial intelligence, where it is desired that the representation be executable. The emphasis here is on modeling the domain in terms of propositions,

scripts, etc., which can be executed under the constraints of several variables in order to simulate a natural or hypothetical system. (See Brachman & Levesque, 1985, for a review of this area).

KRi is the class of interest here, in which key information about the domain is represented in a way so that instructional decisions may be made. Here the emphasis is on categorizing the elements of the domain for the purposes of selecting instructional strategies, and identifying the semantics of links among domain elements in order to prescribe instructional sequences. ID-1 approaches to knowledge representation (referred to as content, or job/task analysis, see Bloom et al., 1956; P.F. Merrill, 1987; Gagné, 1985) are insufficiently precise and comprehensive, and are particularly lacking in describing linkages among domain elements.

Knowledge Representation for ID-2

The key to ID-2 is the acquisition and representation of course content. We propose to represent knowledge in terms of objects which we call frames; each frame has an internal structure (slots, which contain values for the structure), and links to other frames. These (both internal and external) are termed elaborations of the frame. The set of all elaborated frames together, which contains all the knowledge to be instructed by a course, is called an elaborated frame network.

It is hypothesized that there are three fundamental frame types:

entities, which correspond to some thing, for example a device, object, person, creature, place, or symbol;

activities, sets of related actions to be performed by the learner; and

processes, sets of related actions which are entirely external to the learner.

There are also three types of elaborations. These are:

components, which correspond to the internal structure of a frame; for an entity, the components would be parts of the entity; for an activity, steps; and for a process, events and causes;

abstractions, which correspond to a "kinds-of" class/subclass hierarchy into which the frame may be classified; and

associations, which are links to other frames in the network.

The network structure of the knowledge representation allows information to move through the structure, so that data contained in one part of the net affects the data stored elsewhere. Two principal means by which this occurs are:

inheritance, in which attributes of a class or superclass in an abstraction hierarchy are passed to a subclass or instance; and

propagation, in which the contents of a frame influence the contents of another frame connected to it via an association link.

Knowledge acquisition is the process of gathering and organizing all of the information required for the student to acquire a given mental model or set of mental models. The product resulting from knowledge acquisition is an elaborated frame network. Each elaborated frame in this network corresponds to the knowledge required to facilitate the development of a mental model in the cognitive structure of the student.

By representing the organization and elaborations of knowledge structures, it will be possible to select and sequence instructional units which make the structure of the knowledge explicit to the student. However, in order to do so effectively, we need more than just a description of the knowledge structures. We need instructional strategies for teaching integrated wholes, and rules, or prescriptions, for selecting these strategies. In addition, we need larger instructional units, transactions, designed to teach an entire knowledge structure, rather than a single knowledge component.

Transactions

A transaction is defined as a mutual, dynamic, real-time give and take between the instructional system and the student in which there is an exchange of information. The purpose of a transaction is to promote the acquisition of one or more mental models. Instruction designed using ID-2 is in terms of a sequence of these transactions.

The adequacy of a transaction is determined by the degree of active mental processing involved, the completeness with which the transaction promotes acquisition of the target mental model, the degree to which the transaction elaborates a prerequisite mental model, and the degree to which the transaction can be customized to the special needs of individual students or groups of students. Transactions are categorized by the content elements instructed (for example, the component -- parts -- elaboration of an entity frame); and by the instructional strategy implemented for that content.

Frequently the effective implementation of a particular instructional strategy will require more than a single transaction. The set of transactions that implements a given instructional strategy (to teach a particular elaborated frame in a particular domain, in order to promote the acquisition of a given mental model by a given student) is referred to as a transaction frame set. Similarly, the set of transactions necessary to achieve a given instructional goal is referred to as a transaction goal set.

Note that these terms refer to the effects of the transactions, not to collections of code segments. A transaction is not a piece of code or an instructional artifact but the effect of executing a piece of code or an instructional artifact. It is a logical construction, not a physical one.

The code segment which when executed causes a transaction to take place is called a transaction instance. This is created by the transaction generation system (TGS) as a result of an object-oriented design process. A class "transaction" is defined, with a subclass structure beneath it corresponding to the different types of possible transactions that can be generated by the system. A given transaction is instantiated by the TGS from this class hierarchy, and the capabilities of the transaction instance are inherited through the class structure. The instructional artifact thus created may be stored in a library of such artifacts for reuse. For efficiency, the TGS may access the library and recommend instantiated transactions in place of generating new transactions. The ability to assemble courses from such previously prepared standard components will be a major source of economy in the instructional design process using ID-2.

The delivery method for a transaction is not constrained by ID-2. In addition, existing instruction, not created with ID-2, may be categorized, placed in the library, and recommended by the system. In order to be included in the library, it is only necessary that a transaction be describable in terms of its intended instructional outcomes and the type of domain knowledge instructed. Examples of possible transactions range from a teacher-given lecture, through CAI and interactive video presentations, to intelligent tutoring systems.

Pedagogic Prescriptions

Instructional strategy specifies a pedagogy for selecting, sequencing, customizing, and integrating instructional units. Strategy exists at several levels. There is strategy embedded into a transaction that controls the presentation of the transaction. This may be termed micro-strategy. Above this level, there is the strategy which directs the assembling of a set of transactions into a transaction frame set, to instruct a particular elaborated frame. There is the higher-level strategy

which integrates the instruction for a set of elaborated frames, each with its own transaction frame set, into a larger instructional unit which corresponds to an instructional goal. At the highest level there is strategy to integrate all goals into a course. These levels may be termed macro-strategies. In this section we are concerned only with macro-strategies.

The identification of instructional goals is critical to the design of instruction for ID-2. A goal corresponds to some learned capability or performance which the student will attain as a result of the instruction. The achieving of a goal may require the acquisition of one, or a set, of mental models by the learner.

An instructional (macro-)strategy is implemented with one or more transaction frame sets. A transaction frame set, as discussed above, is constructed to include all the transactions necessary to promote the acquisition of a given mental model. The pedagogic strategy determines which of the possible transaction instances should be generated to achieve the instructional goal. Any given frame set will include only a few of the transaction instances which could be generated for that type of knowledge structure. Hence the pedagogic strategy serves to direct and constrain the application of the transactions. In addition, because each transaction requires only certain elements of the knowledge structure, the strategy also directs and constrains the knowledge acquisition to just the portions of the knowledge structure required for the goal. Without such constraint, the knowledge acquisition process would be completely open-ended.

A limitation of ID-1 is the lack of pedagogic prescriptions. ID-2 will contain rules for prescribing instructional strategies, which in turn will prescribe transactions. The prescription of instructional strategies will result from an analysis of the requirements and constraints of a particular instructional situation, and will lead to the identification of instructional goals.

Information gathering is the first requirement of strategy analysis. Relevant information includes an analysis of the application to which the learning will be put, the characteristics of the learner population, and the environmental conditions under which the instruction will be administered.

Using this information about a particular instructional situation, strategy analysis provides both prescriptions and filters for the knowledge acquisition process. The knowledge acquisition process is general, that is, a Knowledge Acquisition System knows about frame components, organization and elaboration but not which of these elements may be appropriate for a given situation. A prescription indicates that a particular goal

requires a given level of abstraction (organization) and certain links between frames (elaboration). A filter indicates that a particular goal does not require certain frame components, certain organizational structures and certain elaborative links. ID-2 would provide rules for selecting prescriptions and filters that correspond to particular kinds of goals. A Strategy Analysis System would guide the user to select a goal type consistent with the course to be developed and would then provide prescriptions and filters which would direct the knowledge acquisition process.

Based on the constrained knowledge structure and information about the students and environment, ID-2 would prescribe sequence rules for ordering the resulting elaborated frames which comprise the knowledge structure (elaborated frame network). A Strategy Analysis System would recommend a course organization consistent with the eventual role of the learners and the particular knowledge to be taught. These rules would take into account the interrelationships between frames in the knowledge structure and the propagation among these frames. Propagation means that information contained in one frame, for a certain goal, must also be included as part of another frame for that goal. For example, if a certain course included an activity frame for "creating a budget", this frame may be linked to an activity frame for "using a spreadsheet". If a step in building a budget is to "identify personnel" then "entering the names of the personnel in the spreadsheet" would be an associated activity. In other words, steps for "using a spreadsheet" frame would propagate to the "building a budget" frame. A Strategy Analysis System would know such propagation rules and use them in building a course sequence.

Finally, the particular transactions and their sequencing necessary to acquire a particular mental model for one student with one set of expectations, abilities, previous preparation and attitudes, may be considerably different from the sequence of interactions needed by a student with a different set of these attributes. ID-2 should include rules relating student attributes with available interactions and their sequence. A Strategy Analysis System would construct appropriate transaction frame sets and configure these for students with different values on these relevant characteristics.

An Open System

A limitation of ID-1 is that there is no means of incorporating fine-grained expertise about teaching and learning, gained from research, and applying this in the design process. An example of this type of expertise would be a set of rules for determining the level of motivation of a student, and prescriptions for adjusting the instruction based on that level. Most knowledge in ID-1 systems is not of this type. To the

extent that such knowledge is incorporated, it is "hard-wired" into the system. There is no means to easily upgrade such knowledge as new findings appear in the literature.

Mini-Experts

In the early days of artificial intelligence research, efforts were directed towards developing a general problem solver, capable of dealing with any situation. The first breakthroughs in artificial reasoning came, however, when the focus shifted to the design of systems limited to a specific, and highly constrained domain.

The typical expert system today contains a large rule base, and an inference engine that applies these rules to available data to reach decisions or to make recommendations. These rule bases, however, are monolithic, and directed towards a single decision or set of decisions. The instructional design process, on the other hand, is not one decision but many different decisions. For this reason, we choose to represent pedagogic expertise in a set of mini-experts, each of which functions relatively independently at different parts of the process, and each of which is responsible for a relatively narrow decision. The theory prescribes the function of these mini-experts. It also provides a means by which the various individual decisions can be coordinated and combined to make the larger decisions involved at various steps in the process of design.

An important aspect of this approach is that it provides a means for opening instructional design systems based upon ID-2. While there is much we do not know about teaching and learning, there is nevertheless a large amount of available data. When a research finding can be expressed as a rule in one of the mini-experts, that knowledge can be incorporated into the system. The system is thus open to new knowledge that is accumulated as a result of research. The development of mini-experts will also help to identify more precisely the knowledge that is currently missing. The mini-experts are the key to the evolution of ID-2. Should ID-2 be successful, we can anticipate that research will be directed towards discovering knowledge upon which prescriptions of specific mini-experts can be based, and toward validating the prescriptions of the mini-experts.

Integration of the Phases of Instructional Design

A critical limitation in the systematic application of ID-1 theories has been the lack of integration of the phases of instructional design. The work in each phase is relatively independent of the work in other phases. When similar data is used across phases, it typically must be translated into another form. This translation process is manual, hence no direct linkage exists among these different representations. Thus

changes made in one phase cannot automatically cause corresponding changes in another. The practicing designer, working to a schedule, will usually maintain up to date only the data for the phase currently worked on, and is reluctant to revisit decisions made at earlier phases. These earlier phases, over time, become outdated and not representative of the actual instruction as developed. Because each phase of design results in a sharpening of focus to smaller and smaller units of instruction, important contextual information is lost when data from earlier phases cannot be manipulated concurrently. This is in no small measure responsible for the shortcomings in developing instruction that teaches integrated goals.

A Single Knowledge Representation

ID-2 resolves this limitation by maintaining a single representation of the data throughout the development process. Changes made in one area automatically flow through to other areas and create corresponding changes. Consistency and completeness checks are facilitated. Also, the designer may more easily return to earlier decisions and observe the effects of changing these without having to redo large portions of the design manually.

In addition, there are close interconnections among the phases. As discussed earlier, the strategy analysis phase directs and constrains both the knowledge acquisition and the authoring of transactions.

An Intelligent Advisor

This integration continues through to the delivery of instruction by means of an on-line advisor program.

The prescriptions made at design time are based on the designer's best estimate of the learner population. During the delivery of instruction, information about the learner, his or her aptitude, specific goals, motivation, familiarity, and other factors, as well as the learner's expressed preferences, may be taken into account to modify those prescriptions.

The advisor would have access to the knowledge base, both for the domain and the pedagogic prescriptions. In addition, it would maintain a student model that contained information about the learner. Using the information gathered about the student, the advisor would adjust design decisions to customize the instruction to more adequately meet the characteristics of the student. The advisor could also engage in a mixed-initiative dialog with the student which would allow the student to participate in this decision-making.

Similarly to the approach to the domain knowledge

representation, we propose to implement in the advisor what would be characterized as a "weak" student model: one that is not capable of simulating the actual state of the student's knowledge, or identifying as a result of such simulation the faulty conceptions, or bugs, in that knowledge. Information about the student would be gathered, categorized, and entered as data for mini-experts responsible for pedagogic strategy decisions such as selection and sequencing of transactions.

Comparison with Other Approaches

We have characterized the solution of ID-2 to the problems previously stated as the development of a theory capable of producing pedagogic prescriptions for integrated learning goals, and being an open system so that research results may be incorporated into the design process in the form of rule-based mini-experts.

The problem of effective instructional development for interactive technologies could be and is approached in other ways. We will examine two classes of alternative approaches.

One major approach is to improve the efficiency by which current instructional design theory and methods are applied, by developing expert systems for advice and guidance of designers (for example, Jones & Massey-Hicks, 1987; Ranker, in press; Gustafson & Reeves, in press). This is a conservative, knowledge engineering approach which focuses on representing existing expertise about instructional design in an expert system. The drawback of this approach is the state of knowledge about instructional design, which we believe is inadequate for the task to which it is put.

Another approach which has received considerable attention is the development of micro-worlds to simulate a domain, and intelligent tutoring systems (Sleeman & Brown, 1982; Wenger, 1987; Polson & Richardson, 1988). These approaches attack the far more difficult problem of creating strong domain and student models capable of executing the knowledge of the domain (KRe). There are several difficulties with these approaches. First is the inherent difficulty of the problem, and the expense of creating these systems. Second is an over-reliance on discovery learning as a means of teaching. Discovery learning (Dewey, 1937; Bruner, Goodnow & Austin, 1967; Papert, 1980) is without question useful, but is not equally desirable in all situations. Important limitations of discovery learning are the additional time that is required, the fidelity of the simulation that is required, and the inability to overcome large gaps in prerequisite knowledge or skills. It is not difficult to imagine situations in which discovery is inappropriate and inefficient. For example, a learner experienced in a related domain may be best served by a simple presentation of the similarities and

differences on critical aspects. A learner with no knowledge of a subject may benefit from an organization of the knowledge to be learned so that a mental model into which further knowledge can be related can begin to be built.

We would contend that the most appropriate instructional strategy is a function of the domain to be instructed, a given learner's knowledge of that domain, and the instructional setting. Discovery learning is one strategy among many; the key from an instructional design point of view is having a basis for knowing when to prescribe discovery, and when to prescribe another method.

Note, however, that an ITS or a micro-world simulation, or another means of discovery learning, can be used as a transaction in ID-2. It would be necessary to describe the ITS or micro-world in terms of the types of domain knowledge instructed, the strategy implemented, and the specific elaborated frames instructed (as these simulations are typically not domain-independent).

III. A THEORETICAL BASIS FOR AIDA (Merrill)

Introduction

In this section I will first attempt to define my concept of theory and theory building so that we have similar meta-expectations about the nature of the task. You may then disagree with my metatheory (theory about theories), but at least you will have a framework to understand my attempt at theory presentation. Second, I will itemize (list) the assumptions, entities and top level principles of the theory presented in the previous section and will attempt to elaborate these theory components a little beyond their presentation in that section.

Metatheory -- A Very Abbreviated Theory of Theories

Theory consists of two primary components: objects (entities) and relationships between these objects. The identification and description of the objects involved might be called descriptive theory. The identification of the relationships between these objects might be called prescriptive theory. When the objects involved in a theory are clearly understood (meaning that the reader can easily identify instances of a particular class of objects) then only prescriptive theory (principles or propositions) is required. However, when a theory attempts to explain the world in a new way then the definition of the objects involved is critical and the theory is meaningless without a careful definition of the entities about which the theory is involved.

In our graduate schools we emphasize methodology aimed at testing propositions (hypotheses) but we almost never teach students how to invent new concepts (that is, identify and describe the objects that the theory will be about). Yet, the identification of the right objects is critical if the theory is to have any validity. When physical science was concerned with fire, water, earth and air little progress was made. When someone postulated atoms and molecules as the components of matter considerable progress was made. The identification of appropriate objects is at least as important as the identification and testing of relations between these objects.

In instructional theory one of our problems may be that we are dealing with the wrong objects, objects that will not enable us to make the kind of progress that we would like. The first step in theory building is to identify those objects which enable more powerful propositions. Instructional theory is weak precisely because the objects we have identified are either ambiguous or inappropriate. To progress this theory needs more appropriate instructional objects about which to build our theory.

Theoretical Interpretation

Instructional Objects

In the previous section we listed several limitations of first generation ID theory (ID-1). Many of these limitations address the objects of which instructional design theory is constructed. This set of limitations identifies instructional objects which we feel are inadequate concepts on which to build instructional theory. In the following table I will list the instructional objects which we think are inadequate and the instructional objects which we feel will enable more powerful second generation theory (ID-2):

INSTRUCTIONAL OBJECTS	
<u>ID-1</u>	<u>ID-2</u>
(Inadequate concepts)	(More adequate concepts)
Individual content components : facts, concepts, principles, procedures	Integrated knowledge/skills: frames, schemata, mental models
Content outlines: hierarchy diagrams	Knowledge structures: knowledge base
Course organization: loosely defined	Course organization: based on content struc.
Small presentation components: definitions, examples, practice	Integrated interactions: transactions, and transaction sets
Passive presentations	Interactive transactions
Basic display elements: rules, examples, etc.	Integrated interactions: transaction sets

Table 1. Comparison of Instructional Objects

An essential step in an adequate second generation instructional design theory is to describe these new instructional objects including their attributes and relationship to other objects in the system. These descriptions constitute instructional design theory.

Instructional design theory must confront two major decisions: what to teach (content), and how to teach (pedagogy). We suggested that ID-2 will have the following capabilities:

- (1) Be capable of analyzing, representing, and guiding instruction to teach integrated sets of knowledge and skills.
- (2) Be capable of producing pedagogic prescriptions for the selection of instructional strategies and the selection and sequencing of instructional transactions.

The first relates to content decisions, the second to pedagogy decisions. These two capabilities incorporate the new instructional objects identified in the limitations. The first includes integrated knowledge/skill sets and knowledge structures represented in a knowledge base. The second includes course organization based on transactions and transaction sets.

Content -- Analyzing and Representing Integrated Goals

Theories start with assumptions, untested axioms, on which the rest of the logical structure will be based. What are our assumptions about the new entities we have identified? What are the propositions that follow from these assumptions?

Assumptions Adopted from Cognitive Science:

* Memory consists of both declarative and procedural knowledge. "Declarative knowledge is knowledge that something is the case, whereas procedural knowledge is knowledge of how to do something." (E. D. Gagné, 1985).

* Memory is organized into integrated sets of declarative and procedural knowledge called mental models.

Principles (Propositions) Adopted from Cognitive Science:

* Organization during learning aids in later retrieval of information.

* Elaborations generated at the time of learning new information can facilitate retrieval.

See E. D. Gagné's Cognitive Psychology of School Learning (1985) for an elaboration of these two principles and the research support for these principles.

ID-2 Content Assumptions: (Assumptions and propositions followed by a double asterisk (**)) are significant modifications of existing ideas or are original contributions from our own research efforts.)

* There are different learning outcomes and different conditions are required to promote each of these different outcomes (Gagné's The Conditions of Learning, 1965, 1985).

* Significant learning outcomes consist of complex human activities which require integrated sets of knowledge and skill. We call such a complex human activity an enterprise. The term enterprise was not used in the previous section. We feel that this instructional object is less ambiguous than the term goal or integrated goal. The cognitive representation of all the knowledge and skill required for engaging in this enterprise is called a mental model. **

Enterprises are characterized by the human activity involved. Different enterprises require different kinds and combinations of knowledge and skill. Hence, different learning activities are required to promote the acquisition of mental models required for different kinds of enterprises. **

ID-2 Content Propositions:

Instruction should focus on helping the learner acquire mental models that enable them to engage in complex human enterprises. **

The acquisition of a mental model by a learner is facilitated by instruction that explicitly organizes and elaborates the knowledge being taught during the instruction. **

The above assumptions and propositions represent our fundamental theory of cognitive organization and knowledge structure. These are fundamental propositions on which ID-2 is built.

ID-2 Content Objects:

In the previous section we tried to begin identifying the objects involved in a theory of knowledge structure. Here we present more careful definitions of these objects. This section is primarily a glossary identifying the key theoretical entities for ID-2. A more complete exposition of each of these entities is required. In most cases they are complex objects with several attributes and methods attached. The purpose of this section is merely to identify these entities and to provide a very brief definition of each. A more complete presentation of the theory

is required to completely present each of these instructional objects but such a presentation is beyond the scope of this section.

Enterprise. An enterprise is some complex human activity. Webster defines enterprise as "...an undertaking that is difficult or complicated..." or "...any systematic or purposeful activity." An enterprise is sometimes described by an integrated goal. An enterprise involves a set of related instructional objectives. An enterprise has a sense of belonging and can be bounded in some way that defines those activities that are part of a particular enterprise and those activities which are not. The terms goal and objective are not technical terms in ID-2 but are used for purposes of clarification.

Enterprise class. A set of enterprises that are characterized by the nature of the human activity involved. Five learned enterprise classes have been identified: denote, manifest, execute, discover, or design. When the term class is used we imply a class hierarchy which may be comprised of several levels of classes and subclasses. For purposes of this section we have identified only the top level of this class hierarchy. For example, the class entity enterprise has as subclasses persons or creatures, places, events, symbols, objects or devices. Similarly, activities and processes have subclasses and these subclasses have further subclasses. A more complete presentation of ID-2 will involve careful definition of these enterprise classes together with all of their attributes. Such a complete presentation is beyond the scope of the current task (see Gagné & Merrill, 1990).

Mental model. The cognitive representation of all the knowledge and skill necessary to carry out an enterprise. A mental model is composed of a set of individual cognitive schema.

Elaborated Frame Network (EFN). A knowledge base (knowledge structure) containing representations of all the knowledge and skill necessary to carry out an enterprise. An EFN is an external representation of the same knowledge and skill which comprise a mental model. We make no claims about how cognitive structure is organized and elaborated, as this is not well understood. We stand on the weaker, and more defensible assumption, that we can analyze the organization and elaborations of knowledge outside the mind, and presume that there is some correspondence between these and the representations in the mind. An EFN is comprised of knowledge frames including their internal structures and the connections (organization and elaboration) among these frames. Abstraction is the class/subclass hierarchy into which the frame may be classified. Association is the linking of frames to other frames in the network.

Frame. Fundamental object for representing knowledge in an

instructional knowledge base (knowledge structure). Each frame has an internal structure (slots which contain values for the structure) which contains the content (knowledge) components represented by the frame. Frames also contain methods which prescribe inheritance and propagation. Inheritance is the method by which attributes of a class or superclass in an abstraction hierarchy are passed to a subclass or instance. Propagation is the method by which the content of one frame influence the content of another frame connected to it via an association link. A frame is the external representation of the knowledge and skill of a single schema in a mental model.

Frame class. A set of frames that are characterized by the nature of the subject matter content involved. Three knowledge structure frame classes have been identified: entity, activity and process.

Content Rules (Principles):

ID-2 consists of two major sets of propositions or rules. Since ID-2 presupposes intelligent tools, the propositions underlying ID-2 are in the form of expert system rules. The rules identified here represent the rules that would comprise a strategy knowledge base for ID-2. When specified these rules comprise two strategy knowledge bases for an intelligent ID Expert system. The first set of rules governs the selection of subject matter content (What to teach?). This first set of rules provides filters and prescriptions for the knowledge acquisition process. The general form of the proposition is as follows:

EFN = F (Enterprise Class and Attributes)

The frames included, the level of abstraction and the elaboration links of an elaborated frame network (EFN) are a function of the class of enterprise involved and the value of the attributes associated with this enterprise class. These rules include those governing inheritance and propagation among frames in the elaborated frame network. These rules together with the slots in the frames themselves identify all of the knowledge and skill. This is then represented in the EFN so that the EFN can promote the mental model associated with the enterprise.

A detailed list of these knowledge structure principles will consist of hundreds of individual rules. It is this level of detailed learning and/or instructional principles that will be necessary to build an AIDA like system. Broad principles will have value only when they are translated into these detailed rules in a strategy knowledge base. Nevertheless, some of the broad based learning principles which will be considered in

building these more detailed rule-based principles follow. This list is representative rather than exhaustive. Furthermore, a detailed explanation of how these very general learning principles relate to the specific entities identified for ID-2 is beyond the scope of the current task.

* The short-term memory principle: The magic number seven plus or minus two represents short-term memory limits. The number of subclasses at any level should be limited to approximately seven or less. The number of steps in an activity at any level, the number of parts associated with an object at any level, etc., should be limited to approximately seven or less. A knowledge acquisition system must know about and promote this "chunking."

* Inheritance principle: All instances within a class inherit (share) all attributes of the class. Instances may have their own unique attributes in addition. Learning the attributes of the class facilitates generalization to instances of the class. It is not necessary for the learner to learn each instance as unique. It is this property of classes that makes abstraction possible and promotes transfer. Transfer occurs when the learner learns the "abstraction model" represented by a class or superclass and then via the inheritance principle makes application to previously unencountered instances of the class. Generalization is the inverse of inheritance. It occurs when a set of individual instances which all share one or more attributes have been learned and the learner then is able to form a class for these individual cases.

* The entity principle: Entities are required for any enterprise. There cannot exist a human enterprise without at least one entity. Therefore, "identifying" an entity is the fundamental enterprise, required as a prerequisite of all other enterprises. Activity and process frames must always have at least one associated entity frame.

* The tool principle: You cannot teach a tool without an application and you cannot execute an application without a tool. Activity frames can be either "application" or "tool" activities. Every application requires at least one tool (thereby prescribing a link between frames) and every tool must be associated with two or more applications.

* The prerequisite principle: New mental models must be built from previously acquired mental models and the schemata which comprise these. New enterprises are constructed from more fundamental previously acquired enterprises.

* The process principle: Underlying any human activity is a process (sometimes not yet known) which provides an "understanding" of the activity. However, knowing the process is

not prerequisite to executing the activity.

Pedagogy -- Transactions

ID-2 Pedagogy Assumption:

* Instructional interactions should be organized around all those activities necessary to promote the acquisition of a particular mental model. **

ID-2 Pedagogy Propositions:

* Integrated interactions which focus on all of the knowledge and skill which comprise a particular elaborated frame network (EFN or knowledge structure) aid the formation of a corresponding mental model and hence enable the learner to acquire the ability to engage in enterprises requiring this mental model.

* There are different classes of transactions required for efficient and effective acquisition of different types of knowledge frames.

The above assumption and propositions represent our fundamental theory of pedagogy. These are fundamental propositions on which ID-2 is built.

ID-2 Pedagogy Objects:

In the previous section we tried to begin the process of identifying the objects involved in a theory of pedagogy. Herein we present more careful definitions of these objects. (A more complete presentation of the theory is needed to completely present each of these instructional objects; such a presentation is beyond the scope of this report.)

Transaction. A particular instructional interaction with a student. A transaction is characterized as a mutual, dynamic, real-time give and take between the instructional system and the student in which there is an exchange of information (Li & Merrill, 1990). A transaction instance is a piece of computer code which, when executed, causes a given transaction to take place. In ID-2 the word transaction is often used to mean transaction instance. A transaction instance corresponds roughly to a segment in ID-1. However, the term segment is not part of the technical vocabulary for ID-2. The content required for a transaction is contained in a given elaborated frame of the knowledge structure. A transaction instance knows what knowledge

it requires for execution.

Transaction class. Different transactions involve different interactions with students. All transactions which require a particular type of interaction are grouped into a transaction class. The specific implementation of this interaction may differ widely depending on the nature of the specific entities, activities or processes involved; depending on the delivery system involved; and depending on the characteristics of the learners. Several transaction classes have been identified including: naming, classifying, predicting, executing, judging, designing, and discovering.

Transaction Frame Set (TFS). A transaction frame set is the specific individual transactions selected from one or more transaction classes which are required to promote the acquisition of a particular instantiated elaborated frame from the knowledge structure. A transaction frame set implements those interactions necessary to promote the acquisition of a particular schema in a particular domain. A transaction frame set corresponds roughly to a lesson in ID-1, however the term lesson is not part of the technical vocabulary of ID-2.

Interaction strategy. A given interaction can be controlled by a student or by a system. The individual content segments can be selected for the student (tutorial) or the student can interact with the content directly (experiential). The interaction can be for the purpose of presenting the information to the student (expository) or for the purpose of allowing student practice or testing the student (inquisitory). The amount and kind of guidance provided to the student can vary.

Interaction strategy class. Interaction strategies classes can be determined on the basis of at least four interaction dimensions: control (learner to system), pedagogic method (tutorial to experiential), mode (expository or inquisitory), and degree of guidance.

Transaction strategy. The possible sequences of individual interactions within a transaction frame set and the decisions as to which transaction should be next for a particular student and when the student should begin the next transaction is called the transaction strategy.

Transaction strategy class. Different instructional functions require different types of transaction strategies. All transaction strategies which perform a similar instructional function are grouped into a transaction strategy class. Several transaction strategy classes have been identified including: overview, remediation, familiarity, basic instruction and assessment.

Transaction enterprise set. All of the transactions necessary to promote the acquisition of a given enterprise are called a transaction enterprise set. A transaction enterprise set is usually comprised of several transaction frame sets. A transaction enterprise set corresponds roughly to a unit in ID-1, however the term unit is not part of the technical vocabulary of ID-2.

Enterprise strategy. An enterprise strategy is the possible sequences of transaction frame sets and the decisions as to which TFS should be next for a particular student and when the student should begin the next TFS.

Enterprise strategy class. Different levels of performance require different types of enterprise strategies. All enterprise strategies which promote a given level of performance are grouped into an enterprise strategy class. Several enterprise strategy classes have been identified including: novice, qualified, and expert.

Pedagogy rules (principles):

Pedagogy (How to teach?) is the second set of rules which guides the selection and configuration of transactions. The general form of the proposition is as follows:

$$\text{TFS} = F (\text{EF}, \text{EA}, \text{SA} \text{ and } \text{GA})$$

The frames included in a transaction frame set (TFS) and the strategy involved for managing traversals through these transactions are a function of the attributes and content included for an elaborated frame (EF). TFS is also a function of environmental attribute values (EA), student attribute values (SA), and enterprise (goal) attribute values (GA).

$$\text{Goal Strategy} = F (\text{EFN}, \text{SA}, \text{ and } \text{EA})$$

The strategy involved for sequencing the transaction frame sets included in the goal frame set and for managing traverse through these TFSs is a function of elaborated frame network attribute values (EFN) as well as student attribute values (SA) and environmental attribute values (EA).

A detailed list of these pedagogy principles will consist of hundreds of individual rules. It is this level of detailed learning and/or instructional principles that will be necessary to build an AIDA like system. Broad principles will have value

only when they are translated into these detailed rules in a strategy knowledge base. Nevertheless, some of the broad based learning principles which will be considered in building these more detailed rule-based pedagogy principles are:

* Role/function principle. Learners enter instruction with different expectations about how they will use the knowledge and skills learned following instruction. These different roles require that the instruction serve different functions. The transactions used must be consistent with these role/function types.

* Principle of least effort. All else being equal learners follow the path of least effort. Hence, learners with low motivation, aptitude, previous experience require more structured transactions; learners with high motivation, aptitude, previous experience prefer less structured transactions.

* Learner control principle. Students do better if they control their own learning hence maximizing learner control should be a meta-objective of instruction. However, students should only be given as much learner control as they can use to their advantage. Poor instructional decisions should lead to less control; good instructional decisions should lead to more control.

* Active learning principle. Amount of learning is a function of amount of relevant mental effort. Transactions should promote "active" rather than "passive" interactions.

* Practice principle. Learners learn what they do. A primary purpose of learning is to provide guided practice in activities as close as possible to the final integrated skilled performance. Transactions must provide opportunities for gradually increasing levels of guided experiential practice.

* Feedback principle. Practice without performance feedback is not practice and promotes very little learning. Learners should have access to performance feedback. However, transactions should guide learners in obtaining intrinsic feedback and gradually eliminate extrinsic feedback.

* Primary presentation principle. Learners learn best when information is represented in all three primary presentation forms: generality, instance and practice. Transactions should include all three primary presentations.

* Guidance principle. In early stages learners benefit from extensive attention focusing information. However, such guidance can be detrimental in later stages of learning. Transactions should provide for the gradual transfer of guidance

from the instructional system to self guidance provided by the learner.

* Representation principle. Learning is improved when information is represented in more than one way. Transactions should provide for multiple representation of ideas. Learning is suppressed when representation is incomplete, i.e., when critical attributes are not adequately represented.

* Matching principle. Learning is improved when instances are carefully matched with non instances; or when correct execution of activities is carefully matched with incorrect execution or activities; or when correct interpretation of processes is carefully matched with common misinterpretations.

* Divergence principle. Learning is improved when a divergent set of instances is presented.

* Elaboration principle. Learning is improved when simple knowledge and skills are elaborated to form complex knowledge and skills; when specific knowledge and skills are elaborated to form abstract knowledge and skills; when static knowledge and skills are elaborated to form dynamic knowledge and skills.

Prologue

These two sections have attempted to make a list of instructional principles relevant to the design of an Advanced Instructional Design Advisor (AIDA). The scope of this task obviously could not require a complete list. Such a list comprises most of the expertise that needs to be included in an AIDA like system, will no doubt require several person-years to complete, and will probably be an on-going process.

In these two sections we have attempted to outline the primary assumptions, objects, and principles which we feel underlie the design of an AIDA type system. To our very brief list could be added hundreds or even thousands of specific principles. We believe that an adequate instructional design theory must, in fact, be built on such detailed principles. A list of very general learning principles will do little to facilitate building an instructional design expert system. Hence, we chose to provide the architecture of a second generation instructional design theory and to identify the principle instructional entities involved and briefly define those entities which we felt were necessary to construct AIDA.

IV. ELABORATION OF THE AIDA CONCEPT (Merrill)

Introduction

Our conception of an "AIDA" like system and the instructional design theory necessary to implement such a system is outlined in these papers:

Merrill, M. D., Li, Z., & Jones, M. K. (1990a). Limitations of first generation instructional design. Educational Technology, 30(1), 7-11.

Merrill, M. D., Li, Z., & Jones, M. K. (1990b). Second generation instructional design (ID-2). Educational Technology, 30(2), 7-14.

Prototype Systems

As mentioned in Section II, we have built two prototypes for an "AIDA" like system. The first, ID EXPERT v1.0, was implemented on a VAX computer using the S.1 expert system shell. This prototype is described in the following paper:

Merrill, M. D. & Li, Z. (1989) An instructional design expert system. Journal of Computer-Based Instruction. 16(3), 95-101.

The second prototype, ID EXPERT v2.0, was implemented on a Macintosh computer using HyperCard for the interface and NEXPERT Object as the expert system shell. This prototype is described in part in the following paper:

Li, Z. & Merrill, M. D. (Submitted for publication) ID Expert v2.0. A desktop instructional design expert system: part 1 instructional design theory and process.

Approach

This section provides an index of variables, parameters and values (values for a given attribute are set off by square brackets []) which may comprise an "AIDA" like system. The organization of this section is keyed to the six block modular representation of the AIDA functional architecture represented in Figure 1 (see Section 1).

This list of variables and parameters is neither exhaustive nor fixed. In our own work this list is continually evolving. As the rules of the system become more specific, we find it

necessary to revisit these attributes and the legal values for these attributes. The list included here should be considered representative of the type of attributes appropriate for a given part of the system rather than an exhaustive list.

The legal values specified should also be considered first cut. For example, for motivation, familiarity, etc., we have indicated high, moderate and low as values. As the rules become more sophisticated, these values will probably be configured on a scale from 1 to 10. Scaling these values becomes more important when the sophistication of the rules moves from IF-THEN rules to rules which cumulate evidence based on certainty factors. We have explored both types of reasoning in earlier prototypes. We find that we start with simple rules and then evolve to more complex rule structures as our experience with a give decision matures.

We have not provided complete definitions for all attributes. Where we have introduced attributes which are unique to our own work we have tried to provide some minimum level of definition or to include a paper which describes the attributes involved in more detail.

The content representation for the proposed knowledge acquisition system and the transactions which will be selected and configured by the executive system represent frames. A detailed description of these frames is beyond the scope of the current assignment. We have tried to provide at least an index of the key technical terms that we use to describe these frames and some of the attributes (slots and values) that comprise these frames. This information is woefully inadequate for adequate communication, but perhaps this outline can provide a guide for further development as the AIDA concept evolves.

Interpretation Notes

In the following six subsections, square brackets ([]) indicate legal values for an attribute. Curly brackets ({}) indicate slots in a frame which can assume some value.

In several of our other publications we have suggested the idea of using miniexperts. For every decision or piece of information requested by the system the system should provide assistance for the user to make this decision. This assistance is in the form of a miniexpert related to this decision. Each miniexpert would gather additional relevant information and then recommend a value for a given attribute. We have not completed very many miniexperts for our prototype systems. These attributes related to motivation illustrate the type of information that may be requested by such a miniexpert.

Information Component

As shown in Figure 1, the information component consists of three components: Audience (students), Environment (instructional setting), and Task (enterprise to be trained). The attributes, parameters, and values associated with this component are represented below.

Audience (students)

Role [consumer, supervisor, technician, problem solver]
Motivation [high, unconcerned, unmotivated]

Some attributes contributing to motivation:

Why instruction [volunteer, required]
Job promotion [yes, unrelated]
Pay increase [yes, unrelated]
Job change [yes, unrelated]
Request job change [yes, no, no job change]
Familiarity [high, moderate, low]
Mastery Level [high, moderate, low]
Ability [high, moderate, low]

Environment (settings)

While recognizing their importance we have not implemented environment attributes in our previous prototype systems. The following list has not been carefully considered and should be considered as merely a suggestion of the type of items which should be considered.

Location [school, on-the-job, remote]
Group size [individual, teams, small group, large group]
Delivery system [individual study, platform, computer,
video, interactive video, etc.]
Budget []
Schedule []
Resources []

Task (enterprise, job)

We do not believe that behavioral objectives or goal lists are the place to start for instructional design. Rather we feel that it is important to identify some bounded complex integrated human activity as the target of the instruction. If this complex human activity is adequately described and classified then the goals and objectives for this enterprise can be derived from the description of the enterprise rather than the other way around.

Furthermore, we feel that it is more natural for a subject matter expert to describe the activities that are to be trained rather than trying to list goals or objectives.

Enterprise class [denote, manifest, interpret, discover, execute, evaluate, design]

The term enterprise was suggested in this paper:

Gagné, R. M. and Merrill, M. D. (1990). Integrative goals for instructional design. Educational Technology Research and Development, 38 (1), 23-30.

We have extended the ideas in this original paper to include the following seven enterprise classes:

For entities:

Denoting -- communicating the identity, describing the form and structure of some entity, activity or process, or some class of entities, activities, or processes.

For activities:

Executing -- performing some activity.

Evaluating -- judging the performance of an activity.

Designing -- devising a novel activity.

For processes:

Manifesting -- making a process evident by showing its phases and sequence.

Interpreting -- analyzing the cause and effect relationships of a process (thus enabling predictions).

Discovering -- bringing to light a new process.

An adequate description of an enterprise will include the identification of other attributes that are necessary in using this information to provide prescriptions and filters for knowledge acquisition and analysis or for selecting and configuring transactions. The generality attribute [specific case or general case] is one such attribute.

Content Component

We are currently working on a Knowledge Acquisition and Analysis System as a component of an IBM Course Development System. We have taken a somewhat different approach to the

representation of knowledge in an instructional knowledge base. Our approach is represented in the following draft papers which are in preparation for publication: Jones, M.K., Li, Z. & Merrill, M. D. (Submitted for publication) Knowledge representation for ID-2: part 1 and part 2.

There are aspects of our work which are not yet complete. These include the following:

Association rules: What are the rules governing necessary and sufficient associations with other frames.

Propagation rules: What are the rules governing the propagation of information from one frame to an associated frame.

Consistency rules: What are the rules for cross checking information to assure that information supplied at one point in the EFN is consistent with information supplied at some other point.

Elaborated Frame Network (EFN)

An elaborated frame network contains all of the content information necessary to teach some human enterprise. The advantage of an elaborated frame network over other forms of representation is that it contains all of the interrelationships necessary to characterize the integrated knowledge (knowledge and skills) necessary for the enterprise.

An elaborated frame network is comprised of a network of associated frames. A frame with its links to other frames identified is called an "elaborated frame." Each frame has three types of elaboration: associations with other frames, membership in an abstraction hierarchy, and components consisting of all the knowledge and skill associated with the frame.

Frame class [entity, activity, process]

Entity class [symbol, object, creature, place]

Activity class [Execute, judge, design, advocate]

Process class [discrete, chained, cyclical, recursive]

Frame Association

Required associations. Certain associations are required by certain enterprise classes. Each enterprise class is characterized by a "minimum frame set," the types of frames minimally necessary to represent the enterprise. An adequate knowledge acquisition system would include the rules for required

associations and use these rules to guide the knowledge acquisition process.

Propagation. The process of propagation operates within the elaborated frame network. Propagation means that information in one knowledge structure (frame) is transferred or affects the knowledge in an associated knowledge structure (frame). For example, an application activity is almost always linked to a tool activity. Each step of the application activity is executed via steps of the tool activity. Propagation would identify those tool steps required to execute each application step. An intelligent knowledge acquisition system would identify the rules of propagation and use them to guide the knowledge acquisition process.

Types of association. Types of association include the following:

[uses/used by,
involves/involved in,
applies/ applied by,
analogy for/ analogy for,
alternative to/ alternative to,
proximal to/proximal to,
interacts with/interacts with,
associated with/associated with]

Frame Abstraction

Abstraction attributes. In addition to the frame components, frames in an abstraction hierarchy require attributes for determining class membership and relationships. Instances can be ordered along one or more dimensions associated with the class to which the instance belong. Members of one class are discriminated from members of a coordinate class on the basis of one or more attributes that are associated with the superclass. A class can also be treated as an instance of a superclass and hence ordered along one or more dimensions associated with the superclass. Superclasses can be classes or instances of a higher superclass for as many levels as necessary.

Inheritance. An important characteristic of an abstraction hierarchy is that instances inherit components from classes, and classes inherit components from superclasses. That is, each instance in a class will have all of the components (parts, paths, episodes) associated the class. There can be exceptions to inheritance for a given instance or a given class.

instance

{dimensions, relative position of each instance
on each dimension }

```

class
  (attributes, value on each attribute which
    determines class membership)
  (dimensions, boundaries for each superclass
  (attributes, legal values for each attribute)

```

Frame Components

A knowledge acquisition system, unlike first generation instructional design analysis procedures, attempts to acquire all of the content required for the instruction to occur. Frame components represent all of the detailed information related to each frame.

```

for entity frame
  (part {name, location, function})
  (properties {name, description, set of legal values})

for activity frame
  (path {steps {action, decision, loop}}
  (action, object, tool, consequence, object)

for process frame
  (episode {event {action, condition, loop}}
    {actor, action, object, consequence, object}
    {causal net}

```

AIDA Executive

The executive is the expert system which takes the information and content and prescribes strategies (transactions) for the student. Two functions are important for an executive function: (1) prescriptions and filters for knowledge acquisition, and (2) selection and configuration of transactions.

Prescriptions and filters

The knowledge acquisition system, as we conceive of it, knows how to acquire knowledge but does not know where to stop. That is, each piece of information indicates to the system how this piece of information could be associated with other pieces of information. A given enterprise, for a given student population, for a given situation, will require only a subset of the possible information which could be acquired. The AIDA executive uses its knowledge about enterprises, students, and situations to send the knowledge acquisition system filters which indicate that a given frame, frame association, abstraction, or

component is not required for a particular instructional implementation. In addition, a given enterprise, student population and/or environment may require some additional frame, frame association, abstraction, or component. The executive will be capable of reminding the user during the knowledge acquisition that a new piece of information is required and to request the necessary content knowledge.

Transaction selection and configuration

The second function of the executive is to use its knowledge about the relationship between student characteristics, environmental characteristics, enterprise characteristics and the nature of the content to select appropriate transactions and configure them for the student. This transaction configuration is what comprises a course with the enterprise transactions determining course organization, providing synthesis and summary functions, providing integrative assessment, etc., and primary transactions providing detailed interactions with the detailed content material.

Strategies Component

Strategies consist of transactions with a student.

Definition

A transaction is the mutual, dynamic, real-time give-and-take between an instructional system and the student in which there is an exchange of information. We distinguish several other terms as well:

A transaction shell is the structure of a transaction identifying the parameters, interactions, content needed, etc. for a given class of transaction. When a transaction shell is instantiated with a particular subject matter for a particular student or group of students, it is called a transaction instance. Both a transaction shell and a transaction instance are pieces of computer code that, when delivered to a student via an appropriate delivery system, cause a transaction to occur. We are not always careful to distinguish the computer objects which cause a transaction to occur from the transaction, the actual interaction with the student.

Our definition of transactions is still very much in progress. The following paper describes the nature of a transaction shell and its role in courseware authoring:

Li, Z. & Merrill, M. D. (1990). Transaction shells: a new approach to courseware authoring. Journal of Research on Computing in Education, 23 (1), 72-86.

Parameters for all Transactions

All transactions share certain parameters which enable the transaction to be configured in a number of ways. Some of these parameters are unique to a particular transaction class but others are shared by all transactions.

mode [expository, inquisitory]

Expository (presentation) is the ability to present the content information to the student. This does not mean there is no interaction but rather that the primary focus of the interaction is for the student to acquire some new information.

Inquisitory means to require the student to demonstrate that they have acquired the desired capability (knowledge or skill).

A given transaction may have several different expository or inquisitory modes. Which of these modes is provided to a given student for a given frame depends on the executive rules for configuring a transaction. The system may determine which mode of the transaction is appropriate at a given point in the instruction or the student can be given control over transaction mode selection. Again the degree of control allowed is determined by the rules governing transaction configuration.

control [learner . . . system]

System control means that the mode, sequence of interactions within a mode, type and amount of guidance is determined for the student.

Learner control means that the mode, sequence of interactions within a mode, type and amount of guidance is determined by the student with or without performance information provided by the system.

method [tutorial . . . experiential]

Tutorial method means that the system selects and portions the content to be presented to the student. Using a tutorial method the system would lead the student carefully through the content. The tutorial method does not imply that the sequence is fixed or rigid. An intelligent tutorial may respond to the student's performance in selecting the next item for presentation or practice. The key characteristic of tutorial method is that the system makes the selection of the next content to be

presented.

Experiential method involves putting the student into a simulation or micro world where the student can interact with the content in a more natural way. The student can explore the domain in a variety of ways and usually has access to a much wider range of content. The experiential environment can be intelligent enabling the student to see what happens if or can be merely a controllable visualization.

display (location, source, timing)

All transactions must be able to accommodate a wide variety of message design parameters. While a given transaction may have a default display configuration this configuration must be able to be modified to accommodate a variety of subject matter representations.

All transactions should be multi-media (this does not imply a delivery system) in the sense that a transaction should be able to accommodate a wide range of display and response characteristics. That is, the display should accommodate text, graphics, audio, and video in any combination. The transaction should also accommodate a wide variety of student response methodologies including constructed responses, pointing, graphing, device simulations, etc.

Enterprise Transactions

An enterprise transaction comprises all of the interactions necessary for a student to acquire the mental model necessary for a given enterprise. An enterprise transaction performs the following functions:

1. Knows all the frames required for a minimum and optimal EFN for the enterprise.
2. Knows which frame should be the next area of study for a given student or group of students and when a given student should switch to this frame. This sequence function is controlled via primary and secondary sequence rules.
3. Knows how to synthesize the entire EFN for the student to enable the student to acquire the "big picture" of the enterprise to be learned.
4. Sends messages to appropriate primary transactions to execute all or part of their methods at appropriate times.

An enterprise transaction is the control structure for the

course or unit level instruction.

minimum and optimal EFN

A minimum EFN is the type and association of frames required by a given enterprise. An optimal EFN is the type and association of all of the frames that may be associated with a given enterprise. A given instructional situation or student population often requires less than the optimal EFN.

primary sequence [encyclopedic, case study, naturalistic]

In our opinion there are three primary sequencing techniques: encyclopedic, case study, and naturalistic.

Encyclopedic sequences involve the presentation of organized, cataloged knowledge. Encyclopedic sequences systematically present each tool, each concept, each activity, each process in a logical manner. This is typically the reference manual or text book sequence. Encyclopedic sequence is the sequence most frequently used in formal schooling.

Case Study sequences involve the presentation of a series of carefully selected examples, scenarios, cases which serve to introduce successively elaborate paths or episodes. Case studies are usually accompanied by a graded secondary sequence. Case studies may be a series of exercises, demonstrations, etc. Case study sequences frequently require propagation transactions.

Naturalistic sequences are characterized as on-the-job. New information is introduced as it is encountered in a "real world" setting. Instruction is often introduced on an "as-needed" basis. There is no gradation of sequence (requires a chronological secondary sequence). Instruction can be solicited by the learner or the learner can be monitored and instruction provided when inadequacies are observed.

secondary sequences (vertical, temporal, abstraction)

Secondary sequences are nested within the primary sequences.

vertical sequence [elaboration (top down), prerequisite
(bottom up), flat]

Vertical sequence refers to the introduction of prerequisite information. An elaboration sequence starts with the simplest complete representation of the activity or process and adds layers of complexity as the instruction progresses. Prerequisite information is introduced on an a-needed basis. A prerequisite sequence presents prerequisite skills first then combines them together into more complex activities.

temporal sequence [chronological, graded]

A chronological sequence is in the order of occurrence or according to order of execution. A graded sequence means that the sequence has been contrived on one or more dimensions.

Dimensions of graded sequence:

familiarity [known . . . unknown]
frequency [most frequent . . . least frequent]
criticality [most critical . . . least critical]

abstraction sequence [concrete to abstract,
abstract to concrete, flat]

Abstraction sequence refers to whether the specific or general case is taught first. This determines the direction of traverse in an abstraction hierarchy.

synthesis interaction {learner control, epitome variation}

Two variations of a synthesis interaction exist at an enterprise transaction level. A learner control transaction would give the student access to the EFN and allow the student to explore the network at will in order to get a picture of the domain.

An epitome variation (based on Reigeluth Elaboration Theory) would provide a systematic view of the network stressing the elaboration of the subject matter in a systematic way rather than merely letting the learner wander.

primary transactions [component, abstraction, association]

A primary transaction comprises all the interactions necessary for a student to acquire the content represented by a frame or a set of related frames. A primary transaction performs the following functions:

1. Knows all the frames required for accomplishing its particular mission (the transaction frame set).
2. Knows which frame or frame component should be the next area of study for a given student or group of students and when a given student should switch to this frame. This sequence function is controlled via primary and secondary sequence rules.
3. Knows how to synthesize the entire transaction frame set for the student.

4. Sends messages to other primary transactions to execute all or part of their methods at appropriate times.

Parameters for all Primary Transactions

The level of performance required by a transaction depends on the level of instruction. Too often we assume basic instruction for naive learners when in the real world much of our learning requires a much less intense learning activity. Instructional level is a parameter which determines the intensity with which the student must interact with the material and the consequent level of performance acquired.

Because a transaction can be either expository or inquisitory the same transaction can serve one or more instructional functions. The mode, level of learner control, and other parameters are determined in part by the function. The same transaction may be activated at different points in the instructional process and serve a different function each time it is activated.

instructional level
[overview,
remediation,
familiarity,
assessment,
basic instruction [LC, SC]]

instructional function
[presentation [LC, SC],
practice,
Ieg assessment,
IG assessment,
remediation]

component transactions [naming, execution, interpreting]

Component transactions enable the learner to acquire all of the components which comprise a given frame. There are three classes of component transactions corresponding to the three types of frames: naming for entity frames, execution for activity frames, and interpreting for process frames.

Naming transactions enable the student to acquire the names, functions, properties, relative location of all the parts which comprise an entity.

Execution transactions enable the student to acquire all of the paths in an activity.

Interpreting transactions enable the student to acquire all

of the episodes in a process.

The term acquire in this context has a range of meanings. Included are: verbal information (Gagné's term), that is, for example, remembering or recognizing the paths in an activity or episodes in a process; being able to actually perform the activity; being able to interpret a process by predicting what will happen in a given situation; and being able to explain what is happening in a given situation. The level of performance required of the student is determined by the executive in configuring the transactions. A component transaction can also apply to a frame at the instance level (the specific case) or at a higher level in an abstraction hierarchy such as a class or super class. In the later case the components being acquired are generalized components which apply in a variety of specific cases.

abstraction transactions [judging, classification/decision, generalization, transfer]

Abstraction transactions enable the learner to acquire skills that require the content from several frames in an abstraction hierarchy.

Judging transactions require a class frame with two or more subordinate instance frames. These frames can be entity, activity or process frames. Judging transactions enable the student to acquire the ability to order the instances of a given class on the basis of some dimension (criterion). The dimensions can be any attribute or combination of attributes. Judging the performance of others as they perform an activity is an example. Ordering a set of entity instances is an example.

Classification/decision transactions require a superclass frame with two or more subordinate class frames each of which have two or more instance frames. These frames can be entity, activity or process frames. Classification transactions enable the student to acquire the ability to sort or identify instances as to class membership. Concept identification is an example. Selecting among alternative activities to accomplish some goal is an example. Editing (selecting the appropriate usage) is an example.

Generalization transactions require a superclass frame with two or more subordinate class frames each of which have two or more instance frames. These frames can be entity, activity or process frames. Generalization transactions enable the student to acquire the ability to combine instances of two or more classes into a more general class. Generalization is the inverse of classification.

Transfer transactions require a superclass frame and one or

more class frames. These frames can be entity, activity or process frames. Transaction transactions enable the student to acquire an abstraction model, that is, a generalized path or generalized episode and apply this to a previously encountered class or instance.

association transactions [propagation, analogy,
substitution, design, discovery]

Association transactions enable the learner to acquire skills that require several different associated frames.

Propagation transactions enable the student to acquire one set of skills in the context of another set of skills. While learning an application activity, the student can simultaneously learn a tool activity for doing the application. While learning a tool, the student can simultaneously learn application activities for the tool. While learning a process, the student can simultaneously learn a method activity for studying or observing the process. While learning a method activity, the student can simultaneously learn the process for which the method was devised.

Analogy transactions enable the student to acquire the paths from one activity by likening it to an analogous activity; or to acquire the episodes in one process by likening it to an analogous process or activity.

Substitution transactions enable the student to learn an alternative activity or process by comparison, elaboration, extension of a previously learned activity or process.

Design transactions enable the student to use given frames in the EFN to invent new activity frames not previously included.

Discovery transactions enable the student to use given frames in the EFN to find new process frames not previously included. Given a method activity the student creates new instances of the application of this method and for each instance identifies a causal network eventually identifying an abstraction model or class process frame for the instances. Discovery transactions enable the student to expand the EFN by adding new frames as the result of creative activity.

Delivery Component

Two characteristics should characterize instruction which is delivered as a result of an AIDA like system. First, the transactions should be independent of any specific delivery system. Second, the instruction should be capable of being adapted to an individual student on-the-fly via an intelligent

advisor system.

System Independent Delivery

Transactions which are prescribed by an "AIDA" like system should be device independent. This means that all of the interactions are captured by a computer program which is capable of expressing the nature of these interactions via a set of generic commands. Each delivery device with which the system interacts would have its unique output driver which is capable of understanding the generic commands from the transaction and interpreting them in terms of the specific characteristics of delivery device. A separate driver would be necessary for each specific delivery device but a single output driver would be able to interpret any of the possible transactions prescribed by the automated design system. We are all familiar with this concept in that we often must configure our computer systems with a driver for a particular printer. Each type of printer must have its own driver capable of interpreting the output from our word processor, spreadsheet, etc.

An Intelligent Advisor

An intelligent advisor is like an "AIDA Executive" that operates on-the-fly while a student is engaged in the actual instruction. An intelligent advisor is not the same as an intelligent tutoring system. An intelligent advisor contains pedagogy rules but not specific information about a particular subject matter. An intelligent advisor monitors a student's performance and on the basis of data gathered concerning the student's performance and interaction with the instruction can provide guidance to the student about what frame should be next and when the student should shift to this frame; or what type of transaction should be next and when the student should shift to this transaction. In addition, the advisor can select, instantiate and configure a new transaction for the student as may be required by the student's performance.

Evaluation Component

Evaluation is of two types: evaluation of student performance and evaluation of system effectiveness.

Student performance

In our concept of instruction we do not separate student evaluation from instruction. Each of the transactions described for the system should have the capability of both expository

and inquisitory modes. An inquisitory mode can be used both for practice or for assessment. Since all of the transactions we envision for such a system are interactive, they must be capable of determining the nature of the student's capability or performance at a given moment in time.

That is, the student must either be provided information about the nature of his/her performance in order to make judgements about the adequacy of their own capability state or performance level; or the system must assess the adequacy of the performance or capability state in order to continue or adjust the interaction until a satisfactory level of learning has been achieved. In other words, assessment of student capability and performance is an integral part of the transaction and advisor system. We do not see the need for a separate student assessment system apart from the components already described.

System Evaluation

An adequate delivery system must have the capability of gathering and interpreting performance data on students as they interact with the system. This data can then be used to evaluate the adequacy of the system itself.

We have entertained the idea of a system that is capable of interpreting this data via its own expert evaluation system and updating itself. Such a self correcting system is difficult to conceive or build. A more feasible approach, at least in the short run, is a system that gathers and interprets information and then in cooperation with an instructional designer provides information which will enable the designer to modify a given rule set or to add or modify a transaction. An adequate system must allow the rules that comprise its expertise to be easily modified and updated as evaluation data becomes available as to the adequacy of the decisions made by the system.

We have suggested that such a system should consist of a set of miniexperts, each of which have knowledge about a particular instructional decision. Further, that these miniexperts should use expert system type representations that easily allow the addition or deletion of rules. In this way the maintenance and continual improvement of such a system should be facilitated.

V. CONCLUSIONS (Spector)

A comparison of the previous three sections with Merrill's other published works on second generation instructional design theory (cf., Merrill, Li, & Jones, 1990b) reveals that Merrill's prescriptions for AIDA have closely paralleled the evolution of his ideas concerning ID EXPERT (Merrill & Li, 1989b). However, The other consultants on the AIDA project have had an influence on Merrill's conceptual thinking and on the final design of AIDA.

For example, Professor Gagné has contributed the notion of an enterprise or complex integrated human activity as the orienting notion for knowledge or training task analysis. Halff has argued that there will be families of enterprises and that instructional strategies will most naturally be grouped around those activities. M. Polson has emphasized the need for cognitive task analysis. O'Neil has emphasized the need for evaluation throughout the process of designing and developing AIDA. Reigeluth has argued that the approach to development should be an elaboration based on simplifying conditions, and this incremental approach has in fact been adopted. Tennyson offered an ambitious notion of an intelligent tutoring system for the domain of instructional design which the group felt was beyond the project's means to support. In addition, Tennyson has argued strongly for the need to update the ISD process to reflect advances both in cognitive science and in interactive technologies. His suggestions were forwarded to HQ ATC/XPCR which has responded with an effort to update the Air Force's ISD policies and procedures. These ideas are documented in the previous five volumes of this series.

It is worth making one final comment about the significance of Merrill's theory of instructional design. Merrill's theory that the world of instructional design divides naturally into entities (abstract or concrete objects), activities (human regulated or controlled procedures), and processes (causal mechanisms independent of human control) is obviously useful when constructing an object-oriented software environment for instructional design. What remains to be seen is whether or not instructional developers view the world in this way. If not, then the issue will be whether they can easily accommodate such a theory. This determination, however, awaits completion of the experimental AIDA (XAIDA).

Merrill and the other consultants are certainly to be credited for making the construction of XAIDA possible. When XAIDA is completed (1994), we will possess the means to rapidly prototype lessons according to a variety of instructional strategies and prescriptions, evaluate the effectiveness of those lessons, and advance instructional design theory accordingly.

REFERENCES

- Ausubel, D. P. (1963). The Psychology of Meaningful Verbal Learning. New York: Grune & Stratton.
- Bloom, B. S. et al. (1956). Taxonomy of Educational Objectives. New York: David McKay.
- Brachman, R. J. & Levesque, H. J. (1985). Readings in Knowledge Representation. San Mateo, CA: Morgan Kaufman.
- Bruner, J. S. (1966). Toward a Theory of Instruction. New York: W. W. Norton.
- Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1967). A Study of Thinking. New York: Science Editions.
- Collins, A. (1987). A sample dialogue based on a theory of inquiry teaching. In C. M. Reigeluth (Ed.), Instructional Theories in Action: Lessons Illustrating Selected Theories and Models. Hillsdale, NJ: Lawrence Erlbaum.
- Collins, A. & Stevens, A. L. (1983). A cognitive theory of inquiry teaching. In C. M. Reigeluth (Ed.), Instructional Design Theories and Models: An Overview of Their Current Status. Hillsdale, NJ: Lawrence Erlbaum.
- Dewey, J. (1937). Experience and Education. New York: McMillan.
- Engelmann, S. & Carnine, D. (1982). Theory of Instruction: Principles and Applications. New York: Irvington.
- Gagné, E. D. (1985). The Cognitive Psychology of School Learning. Boston: Little, Brown, & Co.
- Gagné, R. M. (1965). The Conditions of Learning (1st edition). New York: Holt, Rinehart, & Winston.
- Gagné, R. M. (1985). The Conditions of Learning (4th edition). New York: Holt, Rinehart, & Winston.
- Gagné, R. M. & Merrill, M. D. (1990). Integrative goals for instructional design. ETR&D, 38 (1), 23-30.
- Gropper, G.L. (1983). A behavioral approach to instructional prescription. In C. M. Reigeluth (Ed.), Instructional Design Theories and Models: An Overview of Their Current Status. Hillsdale, NJ: Lawrence Erlbaum.

- Gropper, G.L. (1987). A lesson based on a behavioral approach to instructional design. In C. M. Reigeluth (Ed.), Instructional Theories in Action: Lessons Illustrating Selected Theories and Models. Hillsdale, NJ: Lawrence Erlbaum.
- Gustafson, K. L. & Reeves, T. C. (1990). IDiOM: A platform for a course development expert system. Educational Technology, 30 (3), 19-25.
- Jones, M. & Massey-Hicks, M. (1987). Expert CML: the next generation. Paper presented at the Computer-assisted Learning in Tertiary Education Conference, Sydney Australia, Nov 30-Dec 2.
- Landa, L. N. (1983). The algo-heuristic theory of instruction. In C. M. Reigeluth (Ed.), Instructional Design Theories and Models: An Overview of Their Current Status. Hillsdale, NJ: Lawrence Erlbaum.
- Landa, L. N. (1987). A fragment of a lesson based on the algo-heuristic theory of instruction. In C. M. Reigeluth (Ed.), Instructional Theories in Action: Lessons Illustrating Selected Theories and Models. Hillsdale, NJ: Lawrence Erlbaum.
- Li, Z. & Merrill, M. D. (1990). Transaction shells: A new approach to courseware authoring. Journal of Research on Computing in Education, 23 (1), 72-86.
- Lippert, R. C. (1989). Expert systems: tutors, tools, and tutees. Journal of Computer Based Instruction, 16 (1), 11-19.
- Markle, S. M. (1983). Designs for Instructional Designers. Champaign, IL: Stipes.
- Merrill, M. D. (1983). Component display theory. In C. M. Reigeluth (Ed.), Instructional Design Theories and Models: An Overview of Their Current Status. Hillsdale, NJ: Lawrence Erlbaum.
- Merrill, M. D. (1987a). A lesson based on the component display theory. In C. M. Reigeluth (Ed.), Instructional Theories in Action: Lessons Illustrating Selected Theories and Models. Hillsdale, NJ: Lawrence Erlbaum.
- Merrill, M. D. (1987b). The new component design theory: instructional design for courseware authoring. Instructional Science, 16, 19-34.
- Merrill, M. D. (1987c). An expert system for instructional

design. IEEE Expert, Summer, .25-37.

Merrill, M. D. (1988). Applying component display theory to the design of courseware. In D. H. Jonassen (Ed.), Instructional Designs for Microcomputer Courseware. Hillsdale, NJ: Lawrence Erlbaum.

Merrill, M. D. & Li, Z. (1988). Implementation of an expert system for instructional design (phase 2). Army Research Institute technical report.

Merrill, M. D. & Li, Z. (1989a). Implementation of an expert system for instructional design (phase 3). Army Research Institute technical report.

Merrill, M. D. & Li, Z. (1989b). An instructional design expert system. Journal of Computer-Based Instruction, 16 (3), 95-101.

Merrill, M. D., Li, Z., & Jones, M. K. (1990a). Limitations of first generation instructional design. Educational Technology, 30(1), 7-11.

Merrill, M. D., Li, Z., & Jones, M. K. (1990b). Second generation instructional design (ID-2). Educational Technology, 30(2), 7-14.

Merrill, P. F. (1987). Job and task analysis. In R. M. Gagné (Ed.), Instructional Technology: Foundations. Hillsdale, NJ: Lawrence Erlbaum.

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

Polson, M. C. & Richardson, J. J. (1988). Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum.

Ranker, R. A. (1990). A computer based lesson development advisor. Educational Technology, 30 (30), 46-49.

Reigeluth, C. M. (1983). The elaboration theory of instruction. In C.M. Reigeluth (Ed.), Instructional Design Theories and Models: An Overview of Their Current Status. Hillsdale, NJ: Lawrence Erlbaum.

Reigeluth, C.M. (1987). Lesson blueprints based on the elaboration theory of instruction. In C.M. Reigeluth (Ed.), Instructional Theories in Action: Lessons Illustrating Selected Theories and Models. Hillsdale, NJ: Lawrence Erlbaum.

- Scandura, J. M. (1983). Instructional strategies based on the structural learning theory. In C.M. Reigeluth (Ed.) Instructional Design Theories and Models: An Overview of Their Current Status. Hillsdale, NJ: Lawrence Erlbaum.
- Skinner, B. F. (1953). Science and Human Behavior. New York: Macmillan.
- Skinner, B. F. (1957). Verbal Behavior. New York: Appleton-Century-Crofts.
- Sleeman, D. & Brown, J. S. (1982) Intelligent Tutoring Systems. New York: Academic.
- Spector, J. M. (1990). Design and Developing an Advanced Instructional Design Advisor (AFHRL-TP-90-52). Brooks AFB, TX: Training Systems Division, Air Force Human Resources Laboratory.
- Stevens, G. H. & Scandura, J. M. (1987). A lesson design based on instructional prescriptions from the structural learning theory. In C.M. Reigeluth (Ed.), Instructional Theories in Action: Lessons Illustrating Selected Theories and Models. Hillsdale, NJ: Lawrence Erlbaum.
- Wenger, E. (1987). Artificial Intelligence and Tutoring Systems. Los Altos, CA: Morgan Kaufman.