Instructional Transaction Theory:
An Instructional Design Model based on Knowledge Objects

M. David Merrill and the ID_2 Research Group
Department of Instructional Technology
Utah State University, Logan, UT 84322-2830, USA

Educational Technology 1996, 36(3), 30-37

Introduction

In a series of papers in Educational Technology the ID_2 Research Group at Utah State University has described various parts of Instructional Transaction Theory. In Merrill, Li & Jones (1990) we argued for the need for a second generation of instructional design theory. Merrill, Li & Jones (1991) provided an introduction to Instructional Transaction Theory. Jones, Li & Merrill (1990) described our knowledge representation system, this system was elaborated in Merrill & ID_2 Research Group (1993). Merrill, Li & Jones (1992) described instructional transaction shells and parameters. Merrill, Jones & Li (1992) described different classes of transaction shells. This paper attempts to summarize the theory as a whole by providing a brief description of various components and their interrelationship. The theory is evolving and the authors request the reader's patience with the inconsistencies which may be present in this series of papers.

What is Instructional Design Theory?

Instructional Systems Development (ISD)

Instructional systems development is a set of procedures for systematically designing and developing instructional materials. It has been described in a number of sources (e.g. Dick & Carey, 1990; Gagné, Briggs, & Wager, 1988). ISD is a set of procedural steps. The emphasis is primarily on what to do, rather than on how to do it, or why it works. ISD has many varieties but all involve five basic phases: analysis, design, development, implementation, and evaluation. ISD is not instructional design theory.

Instructional Design Theory

Instructional design theory is a set of prescriptions for determining appropriate instructional strategies to enable learners to acquire instructional goals. ID theory is prescription-based and is founded in learning theory and related disciplines. The emphasis is on what works rather than on the steps to carry out the design and development process. ID theory is sometimes nested within ISD.

The type of ID theory addressed in this paper is based on the Gagné (1965, 1985) assumption that there are different kinds of instructional goals and that different instructional strategies are required in order for the learner to most effectively and efficiently acquire a given kind of instructional goal. All ID theory based on this assumption consists of three components: a

 A version of this paper will appear in a volume on Instructional Design Theory being edited by Sanne Djkstra, Norbert Seel, and Robert D. Tennyson. -- IDTHRYK2.doc descriptive theory of the knowledge and skill[1] to be learned, a descriptive theory of instructional strategies required to promote this learning, and a prescriptive theory that relates knowledge and strategies. Descriptive theory identifies the concepts used to describe either the knowledge to be learned or the strategies to be used to promote this learning. Prescriptive theory consists of if... then ... prescriptions of the form: if [kind of knowledge outcome] then [specific instructional strategy]. That is, if the learner is to acquire a particular kind of knowledge or skill then the instruction must employ the instructional strategy that is appropriate for promoting the acquisition of that kind of knowledge.

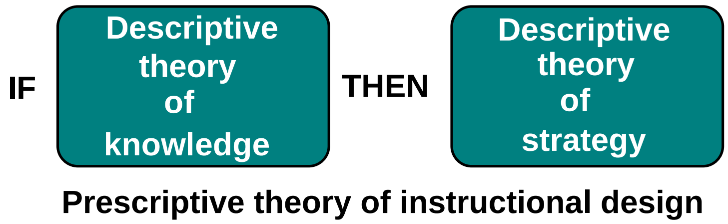

Figure 1 Three components of instructional design theory

Gagné conditions of learning

Gagné (1985) proposed a descriptive theory of knowledge consisting of five outcome categories: intellectual skills, cognitive strategies, verbal information, motor skills, and attitudes. He elaborates some of these categories further. He also proposed a descriptive theory of strategy consisting of nine events of instruction: gaining attention, informing the learner of the objective and activating motivation, stimulating recall of prior knowledge, presenting the stimulus material, providing learning guidance, eliciting performance, providing feedback, assessing performance, and enhancing retention and transfer. For each outcome/event category he then identifies the conditions necessary for learning to be efficient and effective. These conditions of learning comprise his prescriptive theory of instruction.

Merrill component display theory

Merrill (1994) proposed a descriptive theory of knowledge consisting of a two way classification based on performance level and content type. His performance dimension is: remember instance, remember generality, use generality with an unencountered instance, and find a new generality. His content dimension is: facts, concepts, procedures, and principles. Merrill proposed a descriptive theory of strategy consisting of primary presentation forms (PPFs), secondary presentation forms (SPFs), and interdisplay relationships (IDRs). Primary presentation forms consist of: expository generality (rule), expository instance (example), inquisitory generality (recall), and inquisitory instance (practice). Secondary presentation forms consist of information added to facilitate learning such as attention focusing help, mnemonics, and feedback. Interdisplay relationships are sequences involving example-nonexample matching, example divergence, and range of example difficulty. For each performance-content classification, component display theory prescribes the combination of PPFs, SPFs, and IDRs that comprise the most efficient and effective instructional strategy.

---

[1] Some authors make a distinction between knowledge (what is it?) and skill (how to do it?). In this paper we use the single term knowledge to refer to both of these kinds of learning.

Instructional transaction theory

We have tagged both of these ID theories as first generation ID theory. Neither Gagné's conditions of learning nor Merrill's component display theory provide a sufficiently complete set of prescriptions to drive a computer program. Instructional transaction theory is an attempt to extend the conditions of learning and component display theory so that the rules are sufficiently well specified to be able to drive automated instructional design and development.

Figure 2 outlines the descriptive knowledge components, descriptive strategy components and prescriptive rules of Instructional Transaction Theory.

Descriptive Theory Of Knowledge	**Descriptive Theory of Strategy**
• knowledge objects	• transaction shells
• knowledge interrelationships	• conditional parameters

Prescriptive Theory of Instructional Design

- Rules for selecting knowledge objects
- Rules for sequencing knowledge objects
- Rules for selecting instructional transactions
- Rules for sequencing instructional transactions
- Algorithms for enacting instructional transactions
- Rules for changing conditional parameters to adapt instruction to individual learners.

Figure 2 Principal components of Instructional Transaction Theory

Instructional transaction theory describes knowledge in terms of three types of **knowledge objects**: entities, activities, and processes. The descriptive theory of knowledge also identifies **interrelationships** among knowledge objects including: components, properties, abstractions, and associations between entities, activities, and processes.

Instructional transaction theory also identifies a set of instructional algorithms called **transaction shells**. A transaction shell consists of rules for selecting and sequencing knowledge objects. It also consists of a sequence of messages to knowledge objects which cause them to display a multimedia resource representing the knowledge object, display their name or description, change their location, or change their property values and consequently the multimedia resources associated with these changed property values. Instructional transaction theory identifies several classes of instructional transactions including: identification, execution, explanation, judging, classification, generalization, and transfer. Basing instructional transaction theory on knowledge objects enables many of the interactions with learners to include simulations of the devices or phenomena being taught. The prescriptive theory consists of rules for selecting the transaction most effective for promoting learning of a given type of knowledge object or relationship between knowledge objects. The prescriptive theory also consists of rules for adjusting the **parameters** of a given transaction to most effectively promote learning for a student with a given configuration of abilities and aptitudes.

In the remainder of this paper we will provide a brief description of the knowledge representation used for instructional transaction theory. We will also describe the type of rules and messages that comprise an instructional transaction. Finally we will illustrate some of the prescriptive rules that relate knowledge objects and instructional transactions and which are used to adjust the parameters of instructional transactions to meet the learning needs of individual students. This paper describes the theory underlying an instructional design expert system. The authors have implemented many of these ideas in a prototype system, ID Expert™ (See Merrill & ID2 Research Team, in press; Cline & Merrill, in press).

Knowledge Representation and analysis

Jones, Li & Merrill (1990) and Merrill & ID2 Research Team (1993) have described Elaborated Frame Networks as a way to analyze knowledge for representation in an instructional knowledge base. The proposed scheme has proved to be a robust representation system which lends itself to the design and development of instructional design tools for automated development.

Definition of knowledge objects

A knowledge object consists of a set of predefined elements. Each of these elements are instantiated by way of a multimedia resource (text, audio, video, graphic) or a pointer to another knowledge object. Some of these elements include the name of the knowledge object, a portrayal of the knowledge object, the location of the knowledge object's portrayal, and other informational elements such as a description or demonstration.

We have identified four kinds of knowledge objects: entities, activities, processes, and properties. Entities are devices, objects, persons, creatures, places, symbols, things. Activities are actions performed by the learner on some entity. Processes are events that occur in the world that affect some entity often as a consequence of some activity. Properties are qualities or quantities associated with an entity, activity or process.

Knowledge interrelationships

Knowledge objects can be linked via component relationships: an entity can be a part of another entity, an activity can be a step of another activity, or a process can be an event of another process. Such component relationships can have as many levels as necessary.

Knowledge objects can be linked via abstraction relationships. An entity can be an instance or subclass of a parent entity, an activity can be an instance or subclass of a parent activity, or a process can be an instance or subclass of a parent process. Such class-subclass relationships can have as many levels as necessary.

Process, entity, and activity knowledge objects interact in specific ways forming some predetermined relationships called PEAnets. An activity is defined as a change in a property value of an entity by means of a learner's act on a controller. A process is defined as rules for changing the property values of its associated entity or entities. Most processes are conditional on other property values. Hence, when a user acts on a controller it changes a property value, the processes which are conditional on this property value then execute changing other property values. When these property values are changed the display rules associated with the entities which own these properties are activated to change the portrayal of the entity for the student. In this way the representation of the entities are able to simulate changes caused by learner activity and their associated processes.

Elements of knowledge objects

A knowledge object is a container consisting of known slots or elements. Each of these slots is a pointer to a resource (text, audio, video, graphic) or resource configuration (set of multi-media objects) or a pointer to another knowledge object. A knowledge object can identify itself by displaying the resources linked to its name, description, portrayal, portrayal location or demonstration. Table 1 identifies the elements of an entity knowledge object.

Table 1 Formal knowledge representation for entity knowledge object.

Element or slot	**Resource or pointer**
Identification	
Label	text
Name	text
Key words for name	text
Description	text
Key words for description	text
Links to resources:	
Portrayal	resource configuration
Location	screen coordinates of portrayal
Demonstration	resource configuration
Owns components:	
Part 1	pointer to entity object
Part 2	pointer to entity object
Part n	pointer to entity object
Owns properties:	
Portrayal screen location[2]	screen coordinates
Portrayal visible	true/false
Portrayal moveable	true/false[3]
Portrayal sizable	true/false[4]
Property 1	pointer to property object
Property 2	pointer to property object
Property n	pointer to property object
Owns controllers[5] and indicators[6]:	

[2] Screen location becomes a property of the knowledge object. Location is a system property rather than a user defined property.
[3] True means that the learner can move the portrayal of the entity about on the screen.
[4] True means that the learner can change the size of the portrayal of the entity.

controller 1	resource configuration
controller n	resource configuration
indicator 1	resource configuration
indicator n	resource configuration

Owns activities:

Activity 1	pointer to activity object
Activity 2	pointer to activity object
Activity n	pointer to activity object

Owns processes:

Process 1	pointer to process object
Process 2	pointer to process object
Process n	pointer to process object

Guided knowledge acquisition

The EFN knowledge representation facilitates knowledge acquisition from a subject matter expert. The system knows what kind of goals can be associated with knowledge objects. A list of such goals, stated in general terms, is presented to the subject matter expert. The user selects those goals to be included in the lesson. The system asks for the identification of the focus knowledge object. Because the system knows the kinds of links that are possible between entities, activities, and processes it is able to prompt the user to specify the knowledge objects required to complete the instruction that will accomplish the goals specified. For example, the user selects a goal to predict which events will or will not occur under different faulted conditions. The system prompts user to create an entity, attach a process, identify tools, identify controllers and indicators.

Strategy in Instructional Transaction Theory

Algorithms vs. frame-based instruction

Most of the current authoring systems are based on branching programmed instruction and consist of a frame-based approach. The fundamental architecture is to present a display (frame) to the learner and to ask the learner to make a response either to a question or a menu. In most newer systems this display can include multimedia presentations. As a result of this choice the system presents the next frame of information to the learner. The primary instructional strategy is the rules for branching.

---

[5] A controller is a graphic object which enables the learner to manipulate some object by changing its location, changing its state, or changing its value. Controllers can take many forms allowing the student to manipulate the controller by clicking, by entering text or numbers, or by selecting options from a drop-down menu.

[6] An indicator is a graphic object which shows the learner the value of some property. Indicators can take many forms including bar indicators, dials, digital displays, or text displays.

The computer program assumption

Most computer programs, other than education, are based on an algorithm approach rather than a frame-based approach. These computer programs consist of an algorithm plus data. An algorithm is a procedure for performing some symbol manipulation task. Data are the symbols that the algorithm manipulates. Computer algorithms gain their power by being reusable; the program uses the same algorithm over and over with different data.

What is an instructional transaction

Instructional transaction theory is based on this computer program assumption. An instructional transaction shell is a computer program that encapsulates the conditions for teaching a given type of knowledge. We assume that an instructional transaction shell is an algorithm. We assume that we can decouple the subject matter (knowledge) to be taught from the strategy required to teach this knowledge. We assume that the subject matter to be taught is the symbols manipulated by the algorithm and represents the data part of the computer program. Therefore, we can use a given instructional transaction shell (computer algorithm) over and over to teach different knowledge (data). We assume that there can be a number of different kinds of instructional transactions and that the same data (knowledge objects) can be used with different instructional transactions, and that the same instructional transactions can be used with different data (knowledge objects).

From a system point of view a transaction is a set of rules for displaying knowledge objects and elements to a student, and rules for interpreting input from the student. One key to a transaction, that can teach any kind of knowledge, is to identify an appropriate knowledge structure that we can use with a wide variety of subject matters. The second key is to determine a general set of rules for manipulating these knowledge objects and learner input so as to provide a variety of interactions ranging from presentation to assessment.

Uncoupled subject matter

In a frame-based approach to instruction there is a tight coupling between the subject matter to be taught and the strategy used to teach this subject matter. A fundamental assumption of Instructional Transaction Theory is that subject matter is data and that as data it can be uncoupled from the instructional strategy used to teach this subject matter. This means that the subject matter can be specified without consideration for the instructional strategy that will be used to present this subject matter. That the same subject matter can be used in a number of different instructional strategies.

Furthermore, Instructional Transaction Theory assumes that knowledge is a formal representation of the material to be taught (as previously described) and that the information actually presented to the student consists of multimedia resources that can be decoupled from this formal knowledge structure. That is, a given knowledge object has a *portrayal*, the representation of the knowledge object seen by the learner. However, this portrayal is merely an instantiation of an element in the knowledge object, and as such it can easily be replaced without affecting the remainder of the knowledge object or the strategies that may be used to teach the knowledge object. Other information associated with a knowledge object, such as descriptions and demonstrations are also instantiated with multimedia resources and can also be replaced without any other affect on the remainder of the knowledge object or the strategies used to teach this knowledge object.

Architecture for Instructional Transaction Theory

An instructional design expert system based on Instructional Transaction Theory must carry out six important responsibilities: (1) selecting the knowledge objects for instruction, (2) sequencing these knowledge objects, (3) selecting the transactions appropriate for teaching a selected knowledge object or set of knowledge objects, (4) sequencing these transactions, (5) enacting these transactions by conducting the interaction with the student, and (6) adapting the way a given transaction is enacted to meet the needs of an individual student being taught. (See Merrill, Li & Jones, 1992).

Each of these responsibilities is enabled through a set of rules and methods for executing these rules. These rules are conditional and depend on three sources of conditions: information about the elements or relationships in the knowledge base, information about student performance, and parameter values determined either by the course author or by adaptive rules.

Select knowledge objects

Because a knowledge base is uncoupled from the strategies that will be used to teach this knowledge it is possible for several courses to be supported from a single knowledge base. Since the knowledge is decoupled from the strategy, the strategy must first select, from all the knowledge objects which could be taught, those knowledge objects to teach in a given course, lesson, or segment of instruction.

The rules for selecting knowledge objects include rules for dividing the knowledge into mind-sized chunks. A parameter allows a designer to increase or decrease the size of these chunks of information.

A multimedia resource data base which is decoupled from the knowledge means that a given knowledge object may have several different portrayals, descriptions, or demonstrations. The theory includes rules for selecting which of the available resources should be assigned for a given enactment of a transaction. Parameters allow the designer to control this selection.

Knowledge objects have the ability for self explanation, that is given a message to do so, they can display or state their name, description, demonstration, or any of a number of other information elements which a designer may wish to assign to the knowledge object. The theory includes rules for selecting which of the available information messages should be presented in a given course, lesson, segment, or transaction. Parameters enable the user to specify which amplifying information to include.

Automatic objective generation

The EFN knowledge representation scheme makes possible a number of automated instructional design activities. This form of knowledge representation makes possible the generation of objectives based on the knowledge represented in the knowledge base. In operation the user selects a focus knowledge object. Because of its links to associated knowledge objects the knowledge selection rules can generate a number of possible goals that the learner may want to acquire related to this focus knowledge object. The goals are then presented for user selection. For example, the user selects an activity knowledge object named "save a file". Through PEAnet associations this activity knowledge object is linked to an entity named "file" and a process "save a file". This knowledge acquisition system would then display the following possible objectives:

- recognize and name the steps of activity save a file.

- recognize a correct demonstration of each step in the activity save a file.
- predict which events of save a file will or will not occur under different conditions.
- predict which events of save a file will or will not occur under different faulted conditions.

The user then decides which of the possible objectives to include in the lesson being designed. The system then selects the associated knowledge objects from the EFN for inclusion in the course or lesson.

Sequence knowledge objects

Knowledge sequencing involved two strategy decisions: which knowledge object should be taught next? and when is the learner ready to proceed to the next knowledge object? The theory includes rules and parameters for these sequencing decisions.

Automatic knowledge sequencing

The EFN knowledge representation and sequencing rules enable automatic knowledge sequencing. The relationships between knowledge objects specifies the required prerequisite knowledge. For example in order to understand the steps of an activity it is necessary for the learner to know the names of the parts or controllers of the entity involved in carrying out the activity. Hence, a transaction to identity parts is required prior to a transaction to execute an activity. Executing the activity will cause certain processes to be executed, which will in turn cause changes in the portrayal of the entities involved. If the user has selected goals for predicting these underlying processes, a transaction is required to teach the learner how to predict what will happen under different conditions.

Automated sequencing is based on rules for making sequence decisions. The input for these rules are the elements of knowledge objects and their relationships specified in the knowledge base, and the user's selection of the goals to be included in the instruction. Segment assignment rules select the knowledge objects which should be taught in a given segment and specify the preferred order for teaching these knowledge objects. Transaction assignment rules associate which type of interaction is most appropriate for facilitating the learners interaction with a particular knowledge object (see section of this paper on Transaction Selection). When a knowledge object consists of a large number of components, component sequence rules determine which components should be taught first, how many components should be taught in a given interaction, and the sequence through these components. PEAnet sequence rules determine the sequence of transactions for teaching a set of interrelated entity, activity, and process knowledge objects (see section of this paper on Transaction Sequence). Concept sequence rules determine the sequence of transactions for enabling the learner to acquire classification, judging and generalizing skills. Transfer sequence rules determine the sequence of transactions required for a learner to acquire an activity related to one entity, build an abstraction model, and then use the acquired activity with a new entity.

Select transactions

Gagné (Gagné, 1965, 1985) assumed that there are different kinds of instructional goals and that different instructional strategies are required in order for the learner to most effectively and efficiently acquire a given kind of instructional goal. He suggested that each type of knowledge requires unique conditions for learning. Learned performance requires engaging the learner in conditions which are appropriate for a given type of knowledge or skill[7]. Instructional transactions are selected based on the conditions required to acquire the selected goals with the selected knowledge objects.

An instructional transaction is a set of instructional interactions with a learner. It is an algorithm that can be used over and over with different knowledge objects. In addition, an instructional transaction is designed to include all of the interactions (conditions) necessary for the learner to acquire a particular kind of knowledge or skill. The conditions of learning propose different kinds of outcomes each requiring different conditions for learning. Component display theory (Merrill, 1994) proposes different kinds of outcomes each requiring different combinations of primary presentation forms, secondary presentation forms, and different interdisplay relationships. Instructional transaction theory proposes different kinds of instructional transactions each including interaction strategies particularly suited for the promotion of a particular kind of instructional goal.

We have previously suggested that there are a number of classes of transactions (Merrill, Jones, and Li, 1992). Each of these transaction classes can have many subvarieties. Each of these instructional transactions are designed to promote a particular instructional outcome and to provide the conditions of learning most appropriate for the acquisition of the knowledge and skill required.

Component transactions

Component instructional transactions enable the learner to acquire knowledge of the components which comprise a single knowledge object. Component transactions promote acquisition of knowledge that is prerequisite to all other transactions. Many of the skills required in many training situations can be acquired using only these fundamental instructional transactions There are three classes of component transactions: identify, execute, and interpret.

An **identify** transaction helps the learner answer the question: what is an entity? It enables the learner to acquire the names, descriptions, and location of the parts of an entity.

An **execute** transaction helps the learner answer the question: how to do an activity? It enables the learner to learn to recognize, list, and do steps in an activity.

An **interpret** transaction helps the learner answer the question: why a process works? It enables the learner to learn to recognize, list, and predict the events in a process.

Abstraction transactions

Abstraction instructional transactions enable the learner to acquire class, subclass, and instance relationships between knowledge objects. Abstraction transactions promote the ability to transfer or use a skill acquired for one set of instances or subclasses with a new[8] set of instances or subclasses. Abstraction transactions enable the learner to generalize knowledge by acquiring

---

[7] There is a significant body of direct and indirect research supporting Gagne's conditions of learning (e.g. see Merrill, 1994).

[8] New means new to the student. We speak of previously unencountered instances which means that the instances used for practice or assessment are not the same as the instances used for instruction. The learner is able to generalize the skill acquired to a previously unencountered situation.

an abstraction model. An abstraction model is knowledge about the general case of an entity, activity, or process.

A **judge** transaction enables the learner to order new instances of a class. The learner acquires knowledge of dimensional properties and is able to order instances with respect to these dimensional properties.

A **classify** transaction enables the learner to sort new instances into subclasses. The learner acquires knowledge of discriminating properties and is able to sort instances with respect to these discriminating properties.

A **generalize** transaction enables the learner to combine instances into a single class. The learner acquires knowledge of generalizing properties and is able to group instances based on these properties.

A **transfer** transaction enables the learner to apply skills acquired in one situation in a new, but related situation. The learner first acquires the ability to do the steps of an activity with respect to one or more specific entities. The learner then acquires the general form for the steps (abstraction model) and then practices applying these generalized steps in with new entities. For processes, the learner first acquires the ability to make predictions about the events of a process with respect to one or more specific entities (situations). The learner then acquires the general form for the events (abstraction model) and then practices applying these generalized events to predicting events with respect to new entities (situations).

Association transactions

Association instructional transactions enable the learner to acquire important relationships between knowledge objects.

A **decide** transaction enables the learner to select among alternative entities, activities, processes. Decision making involves an association between target PEAnet knowledge objects and decision PEAnet knowledge objects. Changing the property values of the decision PEAnet knowledge objects causes changes in the target PEAnet knowledge objects. The learner acquires the rules that relate the decision knowledge objects with the target knowledge objects and thus is able to select these property values for the decision knowledge objects that result in the desired changes in the target knowledge objects.

A **tool** transaction enables the learner to use one activity to do another activity. A tool is an activity-entity which is used to change the property values of an application process-entity. Effective tool use requires the learner to have knowledge about both the tool and the application. The learner must learn to do the steps of the tool. But the learner must also learn to predict the changes in the application that result from applying and doing the tool.

An **analogy** transaction enables the learner to learn a target PEAnet (process-entity-activity) from a similar original PEAnet. The learner must first have acquired the relevant knowledge with regard to the original PEAnet. The learner must then learn the mapping from the original to the target. The learner must also learn where the analogy stops and where the target goes beyond the analogy.

A **substitute** transaction enables the learner to modify an original process or skill (PEAnet) to acquire a new target process or activity (PEAnet). The learner must first have acquired the relevant knowledge with regard to the original PEAnet. The learner must then learn what is different in the target and where the target goes beyond the original.

A **design** transaction enables the learner to invent a new entity or activity.

A **discover** transaction enables the learner to discover a new process.

Sequence transactions

Rules for sequencing transactions are conditional on student characteristics. Table 2 illustrates two such rules for transaction sequence and interaction sequence within a transaction. These rules use the student characteristics of motivation and experience as conditions. The rules define a variable set of default values for the parameters. *Learner control* means that the learner is given a menu from which to select the next transaction or interaction within a transaction. *Integrated* means that all of one kind of interaction, such as presentation, is enacted for all of the transactions in a segment, before the next type of interaction is enacted for all of the transactions in the segment. *Remedial* means that test or practice interactions are enacted first, and then presentation or exploration interactions are enacted for those topics for which the learner did not meet criterion. *Standard* means that the interactions follow the order: presentation, exploration, practice and assessment. If a given type of interaction is not included in a given transaction it is skipped in the standard sequence.

Table 2 Configuration rule for determining transaction and interaction sequence.

Segment Strategy

Experience	High		Low	
Motivation	High	Low	High	Low
Tx sequence:	learner control	integrated	learner control	integrated
Interaction sequence:	learner control	remedial	learner control	standard

Enact transactions

Table 3a and 3b illustrate some of the rules for enacting an identify transaction. This is only one of a number of different instructional transactions that have been defined for enabling learners to interact with knowledge objects. The rules, as presented here, are pseudo code and must obviously be programmed into a formal programming language to be used in an instructional expert system. The parameters (column 1) and parameter values (column 2) are conditions for the command in the right column to be enacted. If the parameter listed in column 1 has the value listed in column 2 then the transaction carries out the instructions in column 3. Each event element can be selected or deselected independent of other elements. Table 3a are the enactment rules for the presentation interaction of the transaction. Table 3b are the enactment rules for the practice interaction of the transaction. This is not a complete set of rules. We have also designed other types of presentations including explore and assess.

Table 3a Identify transaction presentation mode enactment rules

Parameter	**Value**	**Tx Enactment**
entity elements		
	portrayal	entity displays its resource configuration for portrayal
	location	entity highlights itself to show its location
	name	entity displays its name
	description	entity displays its description
	demonstration	entity displays its resource configuration for demo.
part display	random	display parts in random order
	remove after each	display elements for one part then remove prior to displaying elements for the next part
	simultaneous	display elements for all parts when the learner enters the interaction
presentation order	list of elements	determines the order of presentation for entity elements
element display time*	wait for user input	display element and wait for mouse click before proceeding to the next element
	wait until event ends	display each element and retain until all elements for a given part have been displayed. On mouse click after part remove previous part's elements.
	wait for [ ] secs	display each element for the number of seconds specified
abort presentation	yes	allows the learner to interrupt and stop the presentation

* Applies to each element of the presentation: name text, manifestation, description text, demonstration

Table 3b Identify transaction practice mode enactment rules

Parameter	**Value**	**Tx Enactment**
presentation elements[2]		IDX directs entity to display selected elements.
	portrayal	entity displays resource configuration for entity
	location	entity highlights itself to show location
	name	entity displays its name
	description	entity displays its description
	demonstration	entity displays its demonstration
response elements*		IDX requires the learner to respond by supplying the selected response elements.
	location	learner clicks on the location of the part
	name	learner selects name or types key word for name of part
	description	learner selects description or types key words for description of part

* Each part element can be selected or deselected independent of other elements. Each element can appear on only the presentation list or the response list.

Table 3b continued Identify transaction practice mode enactment rules

Parameter	**Value**	**Tx Enactment**
practice sequencing	simultaneous	requires the learner to complete all of the selected response modes for one part before going on the next part.
	sequential	requires the learner to complete one response mode for all of the parts before going to the next response mode.
	learner control	display a response menu to the learner listing all the selected practice modes.
part order	random	requires the learner to practice parts in random order
	fixed	requires the learner to practice parts in a fixed order
abort practice	yes	allows the learner to abort practice before finishing
mastery	criterion %	the minimum score (% correct) accepted for passing the transaction
	# tries to criterion	number of tries to reach criterion. In standard and remedial segment sequences IDX automatically returns the learner to present or explore if the learner does not reach criterion within this number of tries.
Feedback time	no feedback	no feedback is provided after learner response
	immediate	feedback is provided after each response
	delayed	The system withholds feedback until the learner has completed all the practice interaction for all the parts within the transaction. The system delayed feedback is a final score.
feedback source	system	display IDX default messages
	designer	display user generated feedback messages
feedback type	R/W correct answer	display "right" message followed by correct answer or "wrong" message followed by correct answer
	R/W	display "right" or "wrong" message only
	W only	display "wrong" message only
	CA	display correct answer after right or wrong responses
# tries		number of tries required before displaying a feedback message. Prior to reaching this number IDX displays "try again" message and repeats the practice.
Response timing*	Wait for user input	allows the learner to take as long as necessary to complete the response
	Wait [ ] seconds	display a "time is up" message if the learner does not respond within [ ] seconds
Response mode	recall	requires learner to type key words for name or description
	recognize	requires learner to select name or description from a list

* Separate timing parameters for each of the response elements.

Adapt to individual learners

Once we have identified parameters we can define rules that enable the system to automatically configure these parameters. An instructional design expert system includes rules for automatically configuring transactions given certain student characteristics. Table 4 illustrates a rule for selecting interaction mode based on learner learning level and motivation. This is merely a sample rule and does not represent all of the rules in an instructional design expert system.

Table 4 Configuration rule for selecting interaction mode.

Interaction Mode Selection								
learning level	Overview		Familiar		Basic		Mastery	
motivation	High	Low	High	Low	High	Low	High	Low
present:		selected		selected	selected	selected	selected	selected
explore:	selected		selected		selected		selected	
practice:			selected	selected	selected	selected	selected	selected
assess:							selected	selected

Adaptive rules can be executed at the time a given course is designed or in real-time while a learner is interacting with the course. For example, the system monitors the student's activities for evidence of motivation level. If the monitor function determines that the motivation level has changed, then the system will modify the values of the parameters that are affected by motivation according to rules like those indicated above.

Bibliography

Some of these sources that describe the work of the ID_2 research group were not cited in the paper but are included for reference.

Cline, R. W. & Merrill, M. D. (in press). Automated instructional design via instructional transactions. NATO ASI Series, Volume F 104, *Instructional models for computer-based learning environments.*

Dick, W. & Carey, L. (1990). *The Systematic Design of Instruction*, 3rd Edition. Harper Collins Publishers.

Gagne, R. M. (1965). *The Conditions of Learning*. Holt, Rinehart and Winston.

Gagne, R. M. (1985). *The Conditions of Learning and Theory of Instruction*, 4th Edition. Holt, Rinehart and Winston.

Gagne, R.M., Briggs, L. J. & Wager, W.W. (1988). *Principles of Instructional Design*, 3rd Edition. Holt, Rinehart and Winston.

Jones, M. K., Li, Z., & Merrill, M. D. (1990). Domain knowledge representation for instructional analysis. *Educational Technology*. 30(10), 7-32.

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.

Merrill, M. D., Fletcher, K. A. & Schneider, E. W. (1980). *TICCIT*. Educational Technology Publications.

Merrill, M. D. (1985) Where is the authoring in authoring systems. *Journal of Computer-Based Instruction.* 12(4), 90-96.

Merrill, M. D. (1987). Prescriptions for an authoring system. *Journal of Computer-Based Instruction.* 14(1), 1-8.

Merrill, M. D., Li, Z., & Jones, M. K. (1990a). Limitations of first generation instructional design (ID_1). *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, M. D., Li, Z. & Jones, M. K. (1991). Instructional Transaction Theory: an introduction. *Educational Technology*, 31(6), 7-12.

Merrill, M. D., Li, Z. & Jones, M. K. (1992). Instructional Transaction Theory: responsibilities, methods, and parameters. *Educational Technology*, 32(2), 5-27.

Merrill, M. D., Jones, M. K. & Li, Z. (1992). Instructional Transaction Theory: classes of transactions. *Educational Technology*, 32(6), 12-26.

Merrill, M. D. & ID2 Research Team. (1993). Instructional Transaction Theory: knowledge relationships among processes, entities, and activities. *Educational Technology*, 33(4), 5-16.

Merrill, M. D. with David G. Twitchell (Ed.) (1994). *Instructional Design Theory*. Educational Technology Publications.

Merrill, M. D. & ID2 Research Team. (in press). Automated instructional design and development. *Emerging Computer Technologies in Education* -- papers from ICCE 93.

Merrill, M. D. & ID2 Research Team (in press). Instructional design theory for automated instructional development. *Emerging Computer Technologies in Education* -- papers from ICCE 93.