

# **Instructional Transaction Theory**

## **An Introduction**

**M. David Merrill, Zhongmin Li and Mark K. Jones**

**Utah State University**

**Second Generation Instructional Design Research Project**

Written December 1990

Published Educational Technology, 1991, 31(6), 7-12.

## Instructional Transaction Theory: An Introduction

M. David Merrill, Zhongmin Li &amp; Mark K. Jones

Utah State University

December 1990

**ABSTRACT**

*Instructional transactions* are instructional algorithms, patterns of learner interactions, usually far more complex than a single display and a single response, which have been designed to enable the learner to acquire a certain kind of knowledge or skill. Different kinds of knowledge and skill would require different kinds of transactions. The necessary set of these instructional transactions are designed and programmed once, like other computer applications such as spread sheets. They can then be used with different content topics as long as these topics are of a similar kind of knowledge or skill.

A *transaction shell* is the structure of a transaction identifying the interactions, parameters, and knowledge representation needed for a given class of transactions. A transaction shell is a piece of computer code that, when delivered to a student via an appropriate delivery system, causes a transaction to occur.

A transaction shell consists of two subsystems: an authoring environment and a delivery environment. The users of the authoring environment are subject matter experts and instructors; the users of the delivery environment are students. The authoring environment has a knowledge acquisition system and a transaction configuration system. The knowledge acquisition system queries a subject matter expert concerning the knowledge and skill required by the enterprise. This information is organized and stored in a knowledge base.

The transaction configuration system enables the instructor, or designer, to provide values for a wide range of instructional parameters. These parameters control the nature of the interactions with the learner. Instructional parameters enable a given transaction shell to be customized for a particular student population, learning environment, and learning task.

The instructional delivery environment provides the actual interactions with the student, as enabled by the transactions, controlled by the instructional parameters, and using the knowledge and skills represented in the knowledge base.

**INTRODUCTION**

Interactive instruction (computer-assisted instruction, interactive video) is often an extension of programmed instruction developed in the behavioral psychology era. The emphasis is on shaping the behavior of students. In this programmed instruction based courseware the primary emphasis is well-designed stimuli presented as displays of text or graphics. Often these displays enable only a limited number of interactions between the courseware and students. The most frequently seen interaction is presenting text and graphics, asking a question, evaluating the student's response, providing feedback, and branching to the next display (Merrill, 1985).

Most CBT authoring systems have a frame-based architecture. The primary element is a display (screen) of information consisting of graphics and text. The learner is asked a question, usually consisting of one of the standard question forms of multiple-choice, short answer, true false, matching. Depending on the learner answer, another frame of information is presented. This alternative path procedure is called branching. Frame-based architecture requires that each frame be individually authored and stored in a data base that resembles a file cabinet of displays that can be accessed in various orders depending on the branching structure.

Many noninstructional computer applications have a different architecture, that is, an algorithm plus data. An algorithm is a sequence of computations that can be repeated over and over with different data. A frame-based CAI architecture may be considered as one kind of algorithm, but a very limited algorithm consisting of branching from one display to another. Subject matter content, the knowledge and skills to be learned, can be separated from the instructional interaction with the learner by which these content elements are learned. The instructional interactions are algorithms for interacting with the learner. The subject matter content is data that is used by these instructional algorithms. *Instructional transactions*<sup>1</sup> are instructional algorithms, patterns of learner interactions (usually far more complex than a single display and a single response) which have been designed to enable the learner to acquire a certain kind of knowledge or skill. Different kinds of knowledge and skill would require different kinds of transactions. The necessary set of these instructional transactions are designed and programmed once, like other applications such as spread sheets and word processors. They can then be used with different content topics as long as these topics are of a similar kind of knowledge or skill.

Authoring by way of a *transaction shell* approach consists of selecting those transactions which are appropriate for a given topic and merely supplying the subject matter content in a form that can be used by the transaction. There is no need to determine every display; to determine a branching structure; to select what kind of questions to use, to specify answer processing. Once the transaction shells have been developed they can be used over and over again with no need for additional programming. The cost savings of this algorithm plus data approach to the development of courseware is many times more efficient than a frame-based approach. A one hour lesson that may require 200 or more hours of development using a frame-based approach can be developed in 20 or 30 hours using a transaction shell approach. Furthermore, the frame-based approach may limit the interactions to answering questions whereas a transaction can involve interactive environments which enable the learner to explore the subject matter, manipulate simulations of devices, and other more effective and more complex interactions that are impossible or impractical to build using frame-based systems.

### INSTRUCTIONAL TRANSACTION SHELLS

An *instructional transaction* is a mutual, dynamic, real-time give-and-take between an instructional system and a student in which there is an exchange of information. It is the complete sequence of presentations and reactions necessary for the student to acquire a specific type of instructional goal. It requires active mental effort by the student. Its effectiveness is determined by the match between the nature of the student's interaction and resulting mental processing with

---

<sup>1</sup> We first introduced the idea of an instructional transaction in Merrill 1985, 1987. Subsequently Li & Merrill (1990) described instructional transactions in more detail.

the type of task and subject matter content to be learned.

We subscribe to the notion that learners more easily store cognitive representations of knowledge and skill bundled into interrelated knowledge structures rather than unrelated bits and pieces of information. All of the knowledge and skill required to engage in some complex activity -- such as solving linear equations, driving a car, using an electronic spread sheet -- are highly interrelated and constitute a *mental model*. This complex activity, enabled by a mental model, is called an *enterprise* (See Gagné & Merrill, 1990).

We distinguish several other terms: A *transaction shell*<sup>2</sup> is the structure of a transaction identifying the interactions, parameters, and knowledge representation needed for a given class or family of transactions. When a transaction shell is instantiated with a particular subject matter and with particular values for its parameters, 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 or set of transactions to occur. We are not always careful to distinguish the computer objects which cause a transaction to occur from the transaction which is the actual interaction with the student.

A *transaction class* is a set of similar transaction shells which have similar interaction requirements and similar knowledge representation requirements. A *transaction family* is all of the transactions necessary to enable a learner to acquire all of the knowledge and skill required to engage in a particular enterprise, that is, all of the transactions necessary to enable a learner to acquire a particular mental model. An *enterprise transaction* is a higher level transaction which accomplishes two purposes: first, it functions as a transaction manager, providing the overall direction of the execution of the individual transaction instances in the transaction family. Second, it provides for an integration of the learning facilitated by the individual transactions in the transaction family.

A transaction shell consists of two subsystems: an authoring environment and a delivery environment (see Figure 1). The users of the authoring environment are subject matter experts and instructors; the users of the delivery environment are students. The authoring environment has a knowledge acquisition system and a transaction configuration system. The knowledge acquisition system queries a subject matter expert concerning the knowledge and skill required by the enterprise. This information is organized by the knowledge acquisition system and stored in the knowledge base. The knowledge base represents the knowledge and skill in ways that enable the transactions to use this information in interacting with the learner.

The transaction configuration system enables the instructor, or designer, to provide values for a wide range of instructional parameters. These parameters control the nature of the interactions with the learner. Instructional parameters enable a given transaction shell to be customized for a particular student population, learning environment, and learning task.

The instructional delivery environment provides the actual interactions with the student, as enabled by the transactions, controlled by the instructional parameters, and using the knowledge and skills represented in the knowledge base.

---

<sup>2</sup> To avoid the too often repetition of the entire phrase "instructional transaction" or "instructional transaction shell" we will often adopt the shorter "transaction" or "transaction shell". The reader should understand that the modifier "instructional" is understood in this shorthand usage.

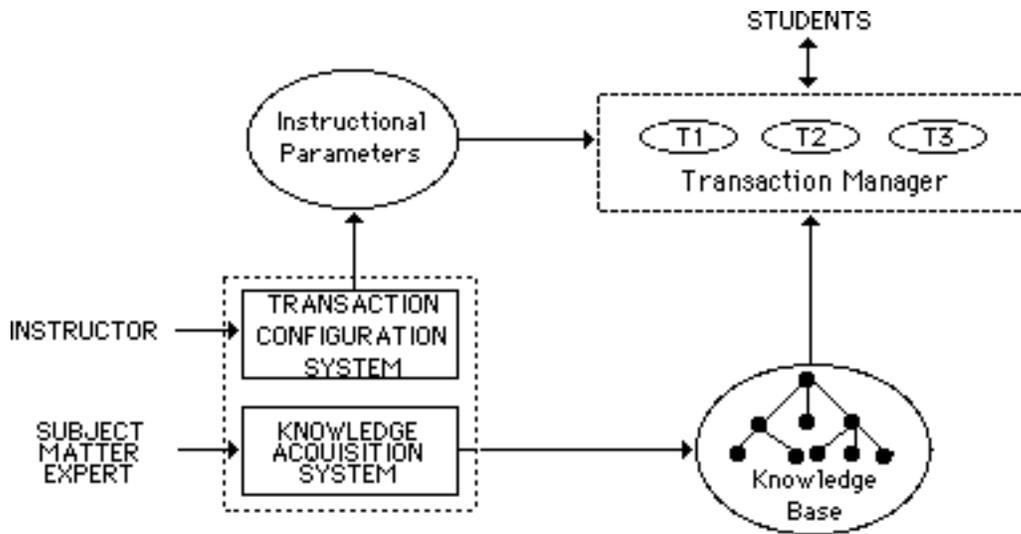


Figure 1 An Instructional Transaction Shell

### COMPONENTS OF AN INSTRUCTIONAL TRANSACTION SHELL

The previous paragraphs and Figure 1 indicates that a transaction shell consists of the following components: a family of transactions with a transaction manager; a knowledge base containing representations of all of the knowledge and skill needed for the transactions; a set of parameters which customize each of the transactions for a given task, learner population, and learning environment; a transaction configuration system; and a knowledge acquisition system.

### TRANSACTION FAMILY

Depending on the complexity of a given enterprise a transaction family can be as simple as a single transaction or it may consist of a large number of transactions all working together to enable the learner to acquire the knowledge and skill associated with the enterprise. A transaction family consists of one or more primary transactions and, when two or more primary transaction transactions are involved, an enterprise transaction.

#### Primary Transactions

A primary transaction is one which is able to promote the acquisition of a particular kind of knowledge or skill. Each kind of primary transaction involves a particular kind of interaction and requires a particular kind of knowledge from the knowledge base. Different classes of primary transactions will be described in a subsequent paper. A transaction family will typically involve instances of several different classes of primary transactions and an enterprise transaction.

A primary transaction can assume a variety of *interaction modes*. These modes enable the transaction to present information, to engage the learner in practice, or to assess the learner's knowledge and skill. An *interaction manager* enables the transaction to sequence the interaction modes as appropriate for a given learner and learning task.

### Enterprise transaction

Primary transaction shells each enable the learner to acquire a particular kind of knowledge and skill, but by themselves cannot promote the integration of the knowledge and skill that is required by the enterprise. This integration must be accomplished by transactions at the enterprise level.

An enterprise transaction accomplishes two principal purposes. First, it functions as a *transaction manager*, providing the overall direction of the execution of the primary transaction instances which instruct the knowledge and skills necessary to the enterprise. Second, it provides for an integration of the learning facilitated by the primary transactions, ideally in the context of a performance or simulation of an authentic activity that is representative of the real-world performance of the enterprise.

This integration is accomplished by focusing the enterprise on a particular performance that is integrative of the elements of the enterprise. The primary transaction that directly instructs the integrative performance becomes the *focus transaction* for the enterprise. Other transactions are introduced to support the performance on the focus transaction. For example, an enterprise in a course on electronics might be interpreting the functioning of circuits. The focus primary transaction would be an instance of an Interpret (process) transaction shell, instructing the functioning of circuits. Supporting primary transactions might include device and symbol identification transactions, concept identification transactions, mathematics execution transactions, fault isolation transactions, interpret transactions for the devices of the various circuits, and interpret transactions for the processes applied by the devices and circuits<sup>3</sup>. Different classes of enterprise transactions and their appropriate focus transactions will be described later in a subsequent paper.

## KNOWLEDGE BASE

Instructional transaction theory is based on several assumptions about the nature of knowledge. First, it is assumed that specific subject matter content can be separated from its underlying structure. Second, it is assumed that subject matter content can be separated from the instructional strategies necessary to teach this content.

### Knowledge structures

The first assumption is implemented via *knowledge structures*. A knowledge structure is a representation framework for a certain type of knowledge. Different types of knowledge have different knowledge structures. This knowledge representation contains slots (variables) for all the knowledge information necessary to carry on instructional interactions with students to enable them to acquire the knowledge and skill contained in the structure. Each knowledge structure contains information about components necessary to its nature. Each knowledge structure knows how it can be linked to other knowledge structures<sup>4</sup>.

---

<sup>3</sup> The various classes of transactions implied by this example will be defined in a subsequent paper.

<sup>4</sup> Jones, Li and Merrill (1990) previously described a knowledge representation system consisting of three classes of knowledge frames -- entity, activity, and process. This system also describes four types of elaboration:

A given transaction shell is associated with a particular kind of knowledge structure or set of associated knowledge structures. The knowledge acquisition function of the transaction shell knows what type of knowledge structures are appropriate for the knowledge that it knows how to teach. The knowledge acquisition function then queries the subject matter expert for the required knowledge and uses its rules about knowledge structure to organize this incoming information in a way consistent with the other components and functions of the transaction shell.

### Transaction Frame Set

The second assumption, that subject matter content can be separated from the instructional strategies necessary to teach this content, is implemented by the identification of different classes of transactions and the configuration of these transactions for a given learning environment and student population. Each of these transaction classes requires a different combination of knowledge frames for their respective knowledge bases. The combination of knowledge frames associated with a given class of transactions is called the *transaction frame set*. The same knowledge structure may occur in the transaction frame set for different classes of transactions. However, the sequencing, interaction modes, guidance and strategy involved in assisting the student to acquire this subject matter will differ depending on the nature of the transaction involved. Thus the nature of the interaction with a given knowledge structure will differ depending on the transaction by way of which this subject matter information is conveyed to the student. Transaction classes and the transaction frame set associated with each of these transaction classes will be described in a subsequent paper.

## PARAMETERS

Each transaction shell is characterized by a set of parameters which enables the shell to be customized for particular learner populations, tasks, and learning environments. A subsequent paper will identify parameters which characterize all transaction shells. In subsequent papers we will also identify instances of these parameters as they apply to different classes of primary and enterprise transactions.

A complete primary transaction shell must include the following classes of parameters: knowledge sequence alternatives, interaction mode alternatives, instructional strategy alternatives, logistic alternatives, and learning environment alternatives. A complete enterprise transaction shell must also include transaction sequence alternatives

A given transaction shell may require several knowledge<sup>5</sup> units or frames from the knowledge base (elaborated frame network). Since learners can only deal with a limited amount of information at any moment in time, it is necessary to sequence these knowledge units. A given knowledge frame may contain a large number of components. These individual components must also be sequenced for the learner. Each transaction shell knows a variety of ways to sequence the subject matter content both between and within knowledge units.

---

attributes, components, abstractions and associations. The entire knowledge base is called an *Elaborated Frame Network (EFN)*.

<sup>5</sup> We have adopted the term *knowledge* to mean the subject matter to be taught.. This term includes both knowledge and skill. When it is important to distinguish between knowledge and skill the text will provide the necessary cues to make the distinction apparent. In the absence of such cues the reader should assume the generic use of the term *knowledge* to mean the subject matter content to be acquired by the student.

Each transaction shell knows a variety of ways to interact with the student. These include ways to present information, as well as ways for the student to practice; ways to overview the knowledge to be taught, as well as ways to assess the knowledge that has been taught .

Each transaction shell includes a variety of strategies for interacting with the students. Strategy refers to the sequencing of various interaction modes and their accompanying logistics.

The logistics of a transaction shell can be configured in a variety of ways which effect the way that it selects, arranges, and enhances its interaction components to best facilitate a given student's acquisition of the knowledge unit being taught.

Each transaction shell operates within a learning environment which enables the student to navigate among the components of the instruction, establishes the display and response characteristics which are common to each interaction, provides necessary learning tools such as note pads, calculators, and glossaries, and provides for enrollment and record keeping for individual students.

### **KNOWLEDGE ACQUISITION SYSTEM**

The knowledge acquisition function is provided by a knowledge acquisition system (see figure 1). It provides a query system to assist the subject matter expert in providing the necessary knowledge for the knowledge base. The knowledge acquisition system includes rules for structuring this knowledge so that it can be used by the delivery function in conducting the interaction with the student. The function of the knowledge acquisition system and the rules involved in knowledge acquisition will be described in a subsequent paper.

### **TRANSACTION CONFIGURATION SYSTEM**

The design guidance function is provided by a transaction configuration system (see figure 1). It provides a query system to assist the instructional designer in selecting the most appropriate sequences, modes, strategies, and logistics for a given group of students, in a particular learning environment, and for a specific learning task. This configuration system provides a query system to assist the instructor (designer) in providing necessary information about the students, environment, and task. This configuration system also contains rules, based on learning and instructional design theory, for using this information to assign values to the instructional parameters of the transaction. These instructional parameters are then used to select among the alternatives provided by the transaction shell so that the interaction is customized for the task to be taught, the particular student population, and the learning environment. The function and rules involved in the Transaction Configuration System will be described in a subsequent paper.

### **IN CONCLUSION**

We previously suggested (Merrill, Li & Jones, 1990a, 1990b) a number of limitations of first generation instructional design (ID<sub>1</sub>) and indicated a new generation of instructional design (ID<sub>2</sub>) is needed to provide more adequate interactive instruction. Merely providing computer-based tools implementing ID<sub>1</sub> is not sufficient. ID<sub>1</sub> must be extended to provide new instructional design theory. This paper is part of a series of papers in which we will describe Instructional Transaction Theory (ITT) as an example of the type of ID<sub>2</sub> theory we feel is needed.

We suggested that ID<sub>2</sub> is best accomplished via instructional transactions. In this paper we have introduced the components of an instructional transaction and transaction shells. One of those components is a transaction configuration system. A subsequent paper in this series will describe the parameters inherited by all instructional transactions. It is these parameters which characterize an instructional transaction as more than merely a computer algorithm. The parameters incorporated into transaction shells enable them to embody what we know about how people learn. Adjusting the values on these learning parameters enable a transaction shell to be adapted to a wide variety of learners and learning environments.

We also suggested that there are different classes of instructional transactions depending on the knowledge structure involved and the learner capability desired with respect to this knowledge. In a subsequent paper we will describe the knowledge, performance, and interactions which characterize each of these several classes of transactions. As instructional transaction theory unfolds future papers will describe in detail each of these classes of instructional transactions.

### REFERENCES

- Gagné, Robert M. & Merrill, M. D. (1990) Integrative goals for instructional design. *Educational Technology Research & Development*. 38(1), 23-30.
- 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. (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<sub>1</sub>). *Educational Technology*, 30(1), 7-11.
- Merrill, M. D., Li, Z., & Jones, M. K. (1990b). Second generation instructional design (ID<sub>2</sub>). *Educational Technology*, 30(2), 7-14.