Instructional Transaction Theory:

Classes of Transactions

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

Utah State University

Second Generation Instructional Design Research Project

Written October 1991

Published Educational Technology, 1992, 32(6), 12-26.

Instructional Transaction Theory:

Classes of Transactions[1]

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

Utah State University

October 1991

Abstract

We propose an instructional transaction class hierarchy. The top object in this hierarchy is the instructional transaction. We assume that different knowledge structures require different types of instructional transactions. We also assume that different transactions promote the acquisition of different types of learner capability. The first level of our instructional transaction class hierarchy is to discriminate transactions on the basis of the type of knowledge structure(s) taught and the learner capability enabled by the transaction.

We have identified three primary classes of transactions: *component* transactions, *abstraction* transactions, and *association* transactions. *Component* transactions require, as a knowledge base, the components from a single frame from the elaborated frame network. *Abstraction* transactions require, as a knowledge base, two or more frames from an abstraction hierarchy. *Association* transactions require, as a knowledge base, two or more associated frames from the Elaborated Frame Network.

There are three classes of component transactions corresponding to the three types of knowledge frames: *identify* for entity frames, *execute* for activity frames, and *interpret* for process frames. An identify transaction requires either an instance or class entity frame. An identify transaction enables the learner to acquire the names, functions, properties, and relative location of all the parts which comprise an entity. An execute transaction requires either an instance or class activity frame. An execute transaction enables the learner to acquire the steps of an activity. An interpret transaction requires either an instance or class process frame. interpret performance enables the learner to acquire the events and causes in a process.

We have identified at least five classes of abstraction transactions: *judge, classify, decide, generalize, and transfer.* A judge transaction requires a class entity, activity, or process frame with two or more subordinate instance frames. A judge transaction enables

[1] This paper reports research in progress. Funding, in part, has been provided by Utah State University, IBM, Human Technology Inc., MediaShare, and several agencies of the U.S.Government. The opinions expressed are those of the authors and not necessarily those of the sponsoring organizations. Please address inquiries concerning this paper to the authors, at the Department of Instructional Technology, Utah State University, Logan, Utah 84322-2830. Telephone (801) 750-2779. Fax (801) 750-2693.

the learner to acquire the ability to order the instances of a given class on the basis of one or more of the dimensional attributes. A classify, decide, generalize, or transfer transaction requires a superclass entity, activity, or process frame with two or more subordinate class frames each of which have two or more instance frames. A classify transaction enables the learner to acquire the ability to sort or classify instances as to class membership. A decide transaction enables the learner to know *when* to select one alternative from another. A generalize transaction enables the learner to acquire the ability to combine instances of two or more classes into a more general class. A transfer transaction enables the learner to acquire an *abstraction model,* that is, a generalized set of steps for an activity , or a generalized set of events for a process, and to apply this abstraction model to a previously unencountered class or instance of the activity or process.

We have identified at least five classes of association transactions: *propagate, analogize, substitute, design,* and *discover.* A propagate transaction enables the learner to acquire one set of skills in the context of another set of skills. An analogize transaction enables the learner to acquire a process or activity by likening it to a different process or activity. A substitute transaction enables the learner to acquire an alternative activity or process by comparison to a previously learned similar activity or process. A design transaction, which is identified but not as completely defined in this paper, enables a learner to invent a new activity or entity. A discover transaction, which is identified but not as completely defined in this paper, enables a learner to find a new process.

Introduction

In previous papers the authors described instructional transactions[2] (Li & Merrill, 1990, Merrill, Li & Jones, 1991, Merrill, Li & Jones, in press). A brief review of your educational experience will convince the reader that there are an enormous variety of ways to interact instructionally with a learner. Are all of these potential instructional transactions unique? Can they be organized in some way that will make an Instructional Transaction Theory manageable? What should be the basis of organizing all of the potential transactions into categories?

In a previous paper we proposed a knowledge representation system. This scheme proposes an *Elaborated Frame Network (EFN)* consisting of three types of knowledge frames: *entities, activities and processes.* We also suggested that these frames could be elaborated in three ways: *component, abstraction,* and *association* elaborations (Jones, Li, and Merrill, 1990). Components consist of parts of an entity, steps of an activity, and events of a process. Abstraction specifies instance, class, and superclass relationships for entity, activity, or process knowledge frames. Associations link frames together on the basis of shared components.

We propose that instructional transactions can be grouped into a limited number of classes. The nature of the interactions for a given class of transaction depends on the type of knowledge structure(s) that the transaction seeks to promote and the learner capability

[2] This paper assumes that the reader is familiar with our concept of instructional transactions.

enabled by the transaction. We assume that different knowledge structures require different types of instructional transactions. We also assume that different transactions promote the acquisition of different types of learner capability.

We have identified three primary classes of transactions: *component* transactions, *abstraction* transactions, and *association* transactions corresponding to the three forms of elaboration in the knowledge representation. The knowledge frames required for a given transaction are called a *transaction frame set.* The transaction frame set for a component transaction is a single knowledge frame and the components which comprise this frame. The transaction frame set for an abstraction transaction is at least a class frame and two or more instance frames from an abstraction hierarchy. The transaction frame set for an association transaction is two or more associated frames from the Elaborated Frame Network.

Component transactions

Component transactions enable the learner to acquire all of the components which comprise a single knowledge frame. The term *acquire* in this context has a range of meanings all the way from *denote,*[3] that is, remembering or recognizing the steps in an activity, or events in a process; to being able to actually *perform* the activity, or *interpret* a process by predicting what will happen in a given situation or explaining what is happening in a given situation. The level of performance required of the learner is a parameter whose value is either specified by the instructional designer or is determined by the authoring configuration or advisor rules of the transaction. A component transaction can apply to a frame at the instance, class, or superclass level. For a class or superclass the components being acquired are generalized components, called an *abstraction model*, which can apply in a variety of specific cases[4].

A component transaction is characterized as follows:

<u>Capability.</u> A component transaction enables the learner to acquire the components of a single knowledge frame -- the parts of an entity, the steps of an activity, or the events of a process.

<u>Knowledge representation.</u> A component transaction requires a single entity, activity, or process knowledge frame at the instance, class, or superclass level.

There are three classes of component transactions corresponding to the three types of knowledge frames: *identify* for entity frames, *execute* for activity frames, and *interpret* for process frames. In the following paragraphs the capability, knowledge required, a brief description of the interactions supported, and associated transactions are described.

[3] See verbal information (Gagné, 1985).

[4] See Abstraction transactions in the next section of this paper.

Identify

Capability

An identify transaction enables the learner to acquire the names, properties, associated information, and relative location of all the parts which comprise an entity. The learner knows *what* it is. Learning the names, location, properties, and associated information of the parts of a entity is prerequisite to learning how an entity works, or how to operate an entity. Learners are shown a physical or functional representation of the entity and asked to identify individual components, properties, associated information, and their immediate connections. Learners are shown both the physical and functional representations and asked to demonstrate the correspondence between these two representations.

Identify transactions have only a denote but no perform level of capability.

Knowledge

The knowledge base for an identify transaction requires either an instance or class entity frame. The component parts, properties, associated information, and their clustering must also be included.

Three representations[5] of the entity are required: a structural representation identifying the clustering of parts and subparts; a physical representation on which the name, location, properties, and associated information for each part can be identified; and/or a functional representation on which the name, location, properties, and associated information for each part can be identified.

Interactions[6]

Information and demonstration level interactions are identical for identify transactions. The shell presents all or a subset of the names to the learner indicating the location, properties, and associated information on one or more of the representations.

Manipulation level interactions require the learner to locate the parts and recognize or provide the name, properties, and/or associated information using one or more of the representations. When symbols are involved the learner may also be directed to reproduce the identifying symbols.

[5] We distinguish the knowledge base, knowledge resources, and knowledge representations. The knowledge base is the formal structural representation of the knowledge. Resources are mediated representations of the knowledge such as text, graphics, video, audio, etc. Representations are the knowledge as it appears to the student and are constructed from the knowledge base and/or appropriate resources. Subsequent papers in this series will describe the relationship between these knowledge components.

[6] In a previous paper (Merrill, Li & Jones, in press) we described four interaction modes: overview, presentation, practice, and assessment. In this paper we use the terms: information, demonstration, and manipulation as the levels of interaction. We feel these terms are more general and plan to adopt them in subsequent papers about Instructional Transaction Theory.

Associated transactions

Identify transactions are usually included in a transaction family involving other transactions (see below) but may stand alone in some situations.

Execute

Capability

An execute transaction enables the learner to acquire the steps of an activity. Execute capabilities can be at either the denote or perform level. At the denote level the learner is able to list the steps involved in a given activity, or shown the activity identify the steps and their sequence. At the perform level the learner knows *how* and is able to do the activity. The learner is shown a physical or functional representation of the entities involved in the activity and asked to do or simulate the steps necessary to manipulate these entities.

Execute capabilities differs depending on the type of entities associated with the activity. Execute capabilities associated with object or device entities include: device *operation* described by verbs such as run, work, manipulate, drive, steer, pilot, play, perform on, use, guide; Device *assembly* or *disassembly* described by verbs such as put together, take apart, fix, repair, collect; device *adjustment* described by verbs such as adjust, test, calibrate, measure; device *trouble shooting* described by verbs such as identify malfunction, detect faults. The device execute capabilities often involves one or more tool entities and activities which must be used to manipulate an application device.

Execute capabilities associated with person or creature entities include: *interact with* described by verbs such as communicate, direct, supervise, control; *perform* characterized by participation in some event.

Execute capabilities associated with symbol entities include: *edit* characterized by detecting and correcting errors in a symbolic communication; *compute* characterized by using arithmetic, logic, or mathematics to manipulate symbols or solve problems; *report* characterized by preparing a form of standardized communication; and *compose* characterized by creating a unique communication.

Knowledge

The knowledge base for an execute transaction requires either an instance or class activity frame. The component steps, substeps, and properties associated with these steps must also be included. The result and/or consequences of each act must also be specified. All activities require one or more associated entities which are the object of the action or the tools by which the action is executed.

Three representations of the activity are required: a structural representation identifying the steps and substeps involved in the activity including the kind of each step (activity, action, condition, or loop), the components of each step (action, object, tool), and the consequent event(s) or state resulting from the action. A dynamic physical and/or functional representation (visualization, manipulation, or simulation[7]) of the entities involved in the activity is also required such that the real or simulated actions performed by the learner result in the illustration of, or simulation of, the resultant consequent events or states resulting from the action.

Interactions.

Information level interactions describe or show the steps involved in the action and the result or consequent of each step. The interaction should enable the learner to recall or recognize the steps, their sequence, and consequences.

Demonstration level interactions demonstrate the activity and the result or consequence of each step in the activity. An adequate demonstration should show different actions which can be taken and the consequence of these different actions. The interaction should enable the learner to recall or recognize the steps and/or the consequences as they are executed.

Manipulation level interactions enable the learner to have some type of hands-on experience with the activity. The learner should be able to do or simulate a step or sequence of steps and see the consequence of their action whether correct or incorrect. The interaction must be able to show the consequence and assess the adequacy of the learner's actions.

Associated transactions

An execute transaction family always involves at least one identify transaction. A learner cannot acquire the steps in an activity unless the entities/parts involved in step can be located and identified. The consequence of an activity is often a change in the parameters of some entity. Many activities require a tool entity in order to execute the steps of the activity.

An execute transaction family may also involve a process transaction. Learners can acquire the ability to execute an activity without knowing the transformations which cause a particular consequence to occur. These transformations are always present and must be understood at some level in order to specify the consequences of the activity. When these transformations and their associated inputs and outputs are taught to the learner then the appropriate process transaction must be included.

[7] We distinguish several levels of dynamic representation. A visualization enables the system to demonstrate the consequence of a given action but may not enable the learner to affect this consequence. A manipulation enables the learner to simulate all or some of the action and observe all or some of the consequence of the action. A simulation is a representation which has a high degree of correspondence with the actual situation such that a change in the action results in an appropriate change in the consequence. Simulations may stress either, or both, physical and functional fidelity.

Interpret

Capability

An interpret transaction enables the learner to acquire the events and causes in a process. interpret capabilities can be at either the denote or perform level. At the denote level the learner is able to list the events involved in a given process, or shown the process identify the events and their sequence. Learners should also be able to identify different states of the process. At the perform level the learner knows *why* it works and can explain the events which lead to a given consequence or can predict the consequence from a series of events. The learner is shown a dynamic physical or functional representation of the entities involved in the process and asked predict what will happen under different normal and/or faulted conditions and/or to identify the conditions upon which a given occurrence of an event in the process was predicated. Prediction includes the ability to recognize when a given process is faulted or not functioning as expected.

Knowledge

The knowledge base for an interpret transaction requires either an instance or class process frame. The component events, and subevents, and properties associated with these events must also be included. The inputs, transformations, and outputs associated with each event must also be specified. All processes require one or more associated entities which are the agents (inputs) which cause the transformation to occur or are the recipients of the transformation (outputs).

Three representations of the process are required: a structural representation identifying the phases, events, and subevents involved in the process including the kind of event (event, episode, condition, or loop), the components of each event (inputs, transformations, outputs) and the causal relationships (and/or relationships of agents) associated with each transformation). A dynamic physical and/or functional visualization, manipulation, or simulation of the entities involved in the process is also required, such that the transformations constituting the events of the process are dynamically illustrated for the learner. These dynamic representations must have the ability to respond to a change in conditions such that the consequent change in transformation for a given event or series of events is illustrated. For manipulations and simulations these dynamic representations must also have the capability for the learner to manipulate the parameters and conditions of the process in order to observe and/or predict changes in the output entities resulting from this change in these parameters or conditions.

Interactions.

Information level interactions describe or show the phases, events, and subevents of the process and the transformations underlying these events. The interaction should enable the learner to recall or recognize the events, their sequence, and statements of the transformations involved.

Demonstration level interactions demonstrate the events and subevents of the process and correlate the underlying transformations with these events. An adequate demonstration should demonstrate alternative events, the conditions leading to these different events, and the underlying transformations associated with these alternatives.

Manipulation level interactions allow the learner to engage in "what if" exploration of the process by adjusting conditions (inputs), observing the consequences (outputs), and observing the differences in the underlying transformations. The transaction must also allow the learner to demonstrate their ability to understand the transformations by predicting subsequent events from a given set of conditions, or identifying the prerequisite conditions that resulted in a given consequence.

Associated transactions

An interpret transaction family always involves at least one identify transaction. A learner cannot acquire the events in a process unless the entities/parts and their properties involved in event can be located and identified. The consequence of a process is often a change in the parameters of some entity.

An interpret transaction family may also involve an activity transaction. Many processes require either a trigger activity to initiate the process or they require a method activity to enable the process to be observed. Learners can acquire the ability to interpret a process without knowing how to execute a trigger activity or use a method activity. These activities are always required and must be understood at some level in order to specify the process. When these trigger or method activities are taught to the learner then the appropriate activity transaction(s) must also be included.

Abstraction transactions.

Abstraction transactions enable the learner to acquire skills that require the content from a class frame and two or more instance frames in an abstraction hierarchy. Abstraction transactions promote the ability to transfer or *use*[8] a skill acquired for one set of instances or classes with a previously unencountered instance or class. Abstraction transactions enable the learner to generalize their knowledge by acquiring an *abstraction model,* knowledge and skills about the general case of an entity, activity, or process.

An abstraction transaction is characterized by:

Capability. An abstraction transaction enables the learner to relate frames from the knowledge base such as ordering a set of instances, categorizing instances as to class membership, identifying instances of subordinate classes as belonging to a superordinate class, selecting among alternative instances, and applying execution or interpretation knowledge or skill attained for one instance to a previously unencountered instance.

Knowledge representation. The knowledge base requires knowledge frames from an abstraction hierarchy for a class of entities, activities, or processes. At least a class and set of instances, or a superclass, a set of coordinate classes, and a set of instances is required.

[2] *Use* is a category of learner performance defined by Component Display Theory. Abstraction Transactions enable a learner to acquire integrated skills that are similar to the component skills identified by the *use* category in CDT (see Merrill, 1983, 1987, 1988).

Abstraction transactions make use of properties of frames or frame components. A property is an attribute, characteristic, feature, or quality of an entity, activity, or process. Properties are usually represented by adjectives or adverbs. They are qualities that modify the frame. They can assume values. All entities, activities, or processes and their components have properties. While properties can be described in isolation they only have meaning when associated with an activity, entity or process or their components. A property can be a component or a property of a component. A property can also be a value (in the sense of emotional value rather than parameter value) attached to an entity, activity or process or their components.

Different types of abstraction transactions can be discriminated on the basis of the performance required and the different combinations of frames from an abstraction hierarchy involved in the transaction. We have identified at least five classes of abstraction transactions: *judge, classify, generalize, decide, and transfer.*

Judge

Capability

A judge transaction enables the learner to acquire the ability to order the instances of a given class on the basis of one or more of the dimensional properties. The learner observes the instances of the entities, activities, or processes. For dynamic properties this observation may involve a demonstration, execution or manipulation of the representation. The learner determines the value of each instance on each relevant property and orders the instances on the basis of these property values.

Knowledge .

The knowledge base for a judge transaction requires an entity, activity, or process class frame with two or more subordinate instance frames. A set of dimensional properties must also be identified for the class frame together with the value and value limits for each dimension. The relative value location of each instance on each dimension must also be included. A property is a quality or quantity associated with an entity, activity or process or a component of an entity, activity or process. A dimensional property is one whose values can be ordered.

Three representations may be required: structural, physical and functional. The structural representation identifies the class frame, each of the subordinate instance frames, and the relevant dimensional properties of the class. If the dimensions for judgement involve dynamic qualities or quantities of the entity, activity or process then these physical and/or functional representations must be able to illustrate these dynamic properties. A dynamic property for an activity is one whose value can only be determined by either observing, or performing, the execution of the activity. A dynamic property for a process is one whose value can only be determined by either observing or manipulating the enactment of the process. When the judgement does not involve dynamic properties then static representations of an activity or process are sufficient.

Interactions.

Information level interactions show the instances and the ordering of the instances.

Demonstration level interactions show the instances, the ordering of the instances, and direct attention to each of the dimensional properties and the values on these properties which were used to determine the ranking of the instances.

Manipulation level interactions require the learner to observe the instances, recognize the dimensional properties, determine a value on each property, and use these values to order the instances. The interaction must provide a mechanism which allows the learner to indicate a ranking for the instances involved.

Associated transactions

A judge transaction family for entities involves at least one identify transaction . The properties used for judgement must be associated with some entity or its parts. If the learner is to be able to identify the value of a particular dimensional property they must be able to find the part associated with the property.

A judge transaction family for activities involves at least one execute transaction and its associated identify transaction. The learner must have at least a denote level of capability with the activity if they are to be able to identify dimensional parameters associated with particular acts that are part of the activity. If execution is required to identify the values for the dimensional parameters then the learner must have a perform level of capability. Acquiring a denote or perform level of capability for an execute transaction requires at least one associated identify transaction as explained above.

A judge transaction family for processes involves at least one interpret transaction and its associated identify transaction. The learner must have at least a denote level of capability with the process if they are to be able to identify dimensional parameters associated with particular events that are part of the process. If interpretation is required to identify the values for the dimensional properties then the learner must have a perform level of capability. Acquiring a denote or perform level of capability for an interpret transaction requires at least one associated identify transaction as explained above.

Classify

Capability

A classify transaction enables the learner to acquire the ability to sort or classify instances as to class membership. It answers the question, What is it? Given an instance from one of the subclasses the learner is able to identify the discriminating properties and the value on each of these properties associated with the instance. For dynamic properties determining the value of a discriminating property may require a demonstration, execution, or manipulation of the representation of the instance. Using this information the learner is able to indicate class membership for the instance.

Knowledge .

The knowledge base for a classify transaction requires a superclass entity, activity, or process frame with two or more subordinate class frames each of which have two or more instance frames. A set of discriminating properties must also be identified for the superclass frame together with the values associated with each property for each subclass. Discriminating properties are those which assume different values for two or more of the subclasses of a given super class. Discriminating properties are usually associated with the frame or components of the frame.

Three representations may be required: structural, physical, and functional. The structural representation identifies the superclass frame, each of the subclasses, representative instances of each of the subclasses, and the discriminating dimensions associated with the superclass and subclasses. If the properties for classification involve dynamic qualities or quantities of the entity, activity or process then these physical and/or functional representations must be able to illustrate these dynamic properties. When the classification does not involve dynamic properties then static representations of an activity or process may be sufficient.

Interactions.

Information level interactions show the classification for the instances.

Demonstration level interactions show the classification of the instances and direct attention to each of the discriminating properties and the values on these properties which were used to determine class membership.

Manipulation level interactions require the learner to observe the instances, recognize the discriminating properties, determine a value for each instance on each property, and use these values to categorize each instance. The interaction must provide a mechanism which allows the learner to indicate the classification for the instances involved.

Associated transactions

A classification transaction family for entities involves at least one identify transaction . The properties used for classification must be associated with some entity or its parts. If the learner is to be able to identify the value of a particular discriminating property they must be able to find the part associated with the property.

A classify transaction family for activities involves at least one execute transaction and its associated identify transaction. The learner must have at least a denote level of capability with the activity if they are to be able to identify discriminating parameters associated with particular acts that are part of the activity. If execution is required to identify the values for the discriminating parameters then the learner must have a perform level of capability. Acquiring a denote or perform level of capability for an execute transaction requires at least one associated identify transaction as explained above.

A classify transaction family for processes involves at least one interpret transaction and its associated identify transaction. The learner must have at least a denote level of capability with the process if they are to be able to identify discriminating parameters associated with particular events that are part of the process. If interpretation is required to identify the values for the dimensional properties then the learner must have a perform level of capability. Acquiring a denote or perform level of capability for an interpret transaction requires at least one associated identify transaction as explained above.

Generalize

Capability

Generalization is the inverse of classification. Generalize transactions enable the learner to acquire the ability to identify what appear to be distinct instances or classes as members of a more general class. Given an instance from one of the subclasses the learner is able to identify the generalizing properties and thus identify the instance as a member of the more general class. Observing dynamic properties may require a demonstration, execution, or manipulation of the representation of the instance.

Knowledge .

The knowledge base for a generalize transaction requires the same knowledge as a classify transaction, that is, a superclass entity, activity, or process frame with two or more subordinate class frames each of which have two or more instance frames. A set of generalizing properties must also be identified for the superclass frame together with the values associated with each property for each subclass. Generalizing, are the same as discriminating properties. They are those properties which assume different values for two or more of the subclasses of a given super class. Generalizing properties are usually associated with the frame or components of the frame.

Three representations may be required: structural, physical, and functional. The structural representation identifies the superclass frame, each of the subclasses, representative instances of each of the subclasses, and the discriminating dimensions associated with the superclass and subclasses. If the properties for generalization involve dynamic qualities or quantities of the entity, activity or process then these physical and/or functional representations must be able to illustrate these dynamic properties. When the generalization does not involve dynamic properties then static representations of an activity or process may be sufficient.

Interactions.

Information level interactions show the several instances and that they can all be grouped together as a more general class.

Demonstration level interactions isolate the generalizing property for each instance, contrast this generalizing property with the same property in a coordinate instance, help the learner see that while the two properties appear different in different instances, that in fact they can be seen as a single abstract property.

Manipulation level interactions require the learner to find the generalizing property in yet other instances and thus recognize these instances as members of the more general class.

Associated transactions

A generalize transaction family for entities involves at least one identify transaction . The properties used for generalization must be associated with some entity or its parts. If the learner is to be able to identify the value of a particular generalizing property they must be able to find the part associated with the property.

A generalize transaction family for activities involves at least one execute transaction and its associated identify transaction. The learner must have at least a denote level of capability with the activity if they are to be able to identify generalizing parameters associated with particular acts that are part of the activity. If execution is required to identify the values for the generalizing parameters then the learner must have a perform level of capability.   Acquiring a denote or perform level of capability for an execute transaction requires at least one associated identify transaction as explained above.

A generalize transaction family for processes involves at least one interpret transaction and its associated identify transaction. The learner must have at least a denote level of capability with the process if they are to be able to identify generalizing parameters associated with particular events that are part of the process. If interpretation is required to identify the values for the generalizing properties then the learner must have a perform level of capability.   Acquiring a denote or perform level of capability for an interpret transaction requires at least one associated identify transaction as explained above.

Decide

Capability

A decide transaction enables the learner to acquire the ability to select one alternative from another. It answers the question, Which do I select? The learner is able identify the effected properties, determine an appropriate value for each of these properties, select a subclass believed to correspond to the desired values of the effected properties, and observe the consequences of a particular choice on the effected properties.

Knowledge .

Decide is discriminated from classification on the basis of the properties involved. Classification involves properties associated primarily with the components of the entities, activities, or processes involved. Decision involves properties associated primarily with a consequent entity, activity, or process rather than the focus frame.

The knowledge base for a decide transaction requires a focus superclass entity, activity, or process frame with two or more subordinate class frames each of which have two or more instance frames.   A decide transaction also requires a consequent superclass entity, activity or process frame associated with the focus superclass frame. A consequent frame is an entity, activity, or process whose properties are effected differently by the different subclasses of the focus frame. A set of effected properties associated with the consequent frame must also be identified together with values on these effected properties that are associated with each of the subclasses of the focus frame. An effected property is one whose value will be different depending on which of the focus subclasses is selected.

Three representations may be required: structural, physical, and functional. The structural representation identifies the focus superclass frame, each of the subclasses, representative instances of each of the subclasses, the associated consequence frame, and the effected properties of the consequence frame that are effected by different choices of subclasses of the focus frame. If the effected properties involve dynamic qualities or quantities of the consequence entity, activity or process then these physical and/or functional representations must be able to illustrate these dynamic properties by showing a consequence of the focus activity or process. When the classification does not involve dynamic properties then static representations of the focus activity or process may be sufficient.

Interactions.

Information level interactions show the focus frame alternative which was selected and the effected properties of the consequent frame.

Demonstration level interactions show the alternative class or instance which was selected. It directs the learner's attention to the consequence class and the properties of the consequence class that are effected by the decision. Different decisions are illustrated together with the different values on the effected properties which correspond to these decisions.

Manipulation level interactions allow the leaner to indicate their choice and to see the consequence of their choice. When consequences are the result of executing activities or manipulating processes then the learner should be able to engage in these executions or manipulations in order to see the consequence of their choice.

Associated transactions

A decision transaction family involves at least an identify, execute, or interpret transaction for the consequence frame. The effected properties used for decision must be associated with some entity and its parts, some activity and its steps, or some process and its events. If the learner is to be able to identify the value of a particular effected property they must have at least a denote level of capability and be able to find the part, step, or event associated with the property.

If the focus frame is an activity which the learner must execute in order to carry out the decision, then the decision transaction family must also include an execute transaction and its associated identify transaction for the focus frame. If the focus frame is a process which the learner must interpret in order to carry out the decision, then the decision transaction family must also include an interpret transaction and its associated identify transaction for the focus frame.

Transfer

Capability

A transfer transaction enables the learner to acquire an *abstraction model,* that is, a generalized set of steps for an activity , or a generalized set of events for a process, and to apply this abstraction model to a previously unencountered class or instance of the activity or process. Transfer enables the learner to apply skills acquired in one situation to a new, but related, situation.

Classification, generalization and decide transactions all involve the psychological process of transfer. A transfer transaction is discriminated from a classification, generalization, or decide transactions by the nature of knowledge that is transferred. Classification, generalization, and decide involve properties of the focus frame, or in the case of decide, the consequence frame. In the case of classify and generalize transactions these properties are used to identify a previously unencountered instance or class. In the case of a decide transactions these properties are used to select among alternative classes or instances. The transfer transaction involves acquiring the steps of an activity, or the events of a process in a specific instance; generalizing these steps or events to build an abstraction model at the class level; and then using these generalized steps or events to execute or interpret a previously unencountered instance of the activity or process.

Knowledge .

The knowledge base for a transfer transaction requires a superclass activity, or process frame, two or more class frames, each with two or more instance frames. The component steps or events must also be included at the superclass, class and instance level. The steps or events at the superclass level comprise an abstraction model which will enable the learner to apply this model to a new class.

Three representations of the activity or process may be required for each frame in the knowledge base: a structural representation, functional representation, and physical representation. The structural representation identifies the superclass frame which represents the generalized knowledge to be transferred, together with a complete specification of the steps or events at the superclass level (the abstraction model). It must also identify each of the class and instance frames together with a complete specification of the steps or events for each of these frames.

The functional and physical representations require a dynamic visualization, manipulation, or simulation of the entities involved in the activity or process. These dynamic representations must have the ability to respond to a user action or to a change in conditions such that the consequent change in the output entities is illustrated. Each of these representations must have all of the functionality of the representations for execute and interpret transaction shells. These can be the same representations as those required for execute and interpret transaction shells except that there must be a set of such representations for all the instances, classes, and superclass involved in the transfer.

Interactions

The primary purpose of the transfer transaction is to help the learner acquire an abstraction model at the class or superclass level and then use this abstraction model with a new instance, or class of instances. The transfer transaction accomplishes this by directing execute or interpret transactions to do their work for instance frames, then for class or superclass frames, and then back to instance frames.

The transfer transaction chooses one instance, directs the execute or interpret transaction to teach the learner the steps or events of the instance. Then the transfer transaction selects a second instance and directs the execute or interpret transaction to teach the learner the steps or events of this instance. It then selects the class frame which is parent to the two instances taught. The transaction helps the learner build the abstraction model, that is, the transaction helps the learner see that the steps or events of the instances are instances of the more general steps or events of the class. The transaction then selects yet another instance and directs the learner to apply the knowledge in this new situation at the perform level.

An information level interaction presents the abstraction model to the learner and illustrates the abstraction model with the steps or events for one or more instances.

A demonstration level interaction directs execute or interpret transactions to teach specific instance or class level step or event components. It also shows the learner how the instance level steps or events are instances of the class level steps or events.

A manipulation level interaction directs an execute or interpret transaction to enable the learner to manipulate an instance. It also enables the learner to jump back and forth to the abstraction model while executing or interpreting a particular instance.

The transfer transaction involves a succession of demonstration and manipulation levels of interaction. It involves a strategy which directs execute or interpret transactions to demonstrate and enable manipulation for the instances and classes of the knowledge structure. It helps the learner to generalize the abstraction model to the specific instances.

Associated transactions

A transfer transaction family involves either an interpret or execute transaction and its associated identify frame. The transfer transaction directs these transactions to successively teach the frames of the transfer knowledge structure. These interpret or execute transactions in turn may call on associated transactions as previously described.

Association transactions

Association transactions enable the learner to acquire skills that require several different associated frames. Association transactions promote the ability to integrate information from two or more knowledge frames into a coordinated set of knowledge and skill. Association transactions enable the learner to use a mental model already acquired to build a modified or new mental model. Association transactions enable the learner to acquire alternative ways to accomplish a given goal. Association transactions enable the learner to invent new entities or activities or to discover new processes.

An association transaction is characterized by:

Capability. An association transaction enables the learner to interrelate knowledge from associated frames such as learning one set of skills in the context of another set of skills, learning by analogy, learning a new skill by reference to a previously learned skill, inventing new entities or properties, or discovering new processes.

Knowledge representation. The knowledge base requires knowledge frames which are associated by relations in which knowledge from one frame propagates to the associated frame or resembles the associated frame in function or appearance.

We have identified at least five classes of association transactions: *propagate, analogize, substitute, design,* and *discover.*

Propagate

Propagation is a process which is embedded in an Elaborated Frame Network (EFN). Whenever two or more frames are associated there is propagation of information between these associated frames. The most basic form of propagation in an EFN is propagation among the knowledge frames of a P.E.A. net (process, entity, activity network). As indicated in previous paragraphs of this paper every activity has at least one associated entity frame; every process has at least one associated entity frame; and every activity has an underlying process which determines the consequence of its acts. Every part or entity which is the recipient of an act in an activity frame must also appear as part of a cluster in an entity frame. Every part or entity whose properties are changed by the transformation of a process must also appear in the parts cluster of an entity frame. This shared information is said to propagate among the associated frames.

It is not necessary to have a separate transaction shell to promote the learning of P.E.A. net propagation. This learning occurs from the enactment of the three component transaction shells: identify, execute, and interpret. Other types of associated frames require a deliberate attempt to promote the acquisition of the relationships between the knowledge of the linked knowledge frames.

Capability.

A propagation transaction makes a deliberate effort to facilitate the learner's integration of information from two or more associated knowledge frames. Propagation enables the learner to acquire one set of skills in the context of another set of skills.

Two primary types of propagation have been identified: tool-application propagation and method-process propagation. A tool is an activity and associated entities for carrying out some other activity and associated entities (the application). A tool is characterized by the statement: "act on entity by tool-act on tool-entity." Tool-application propagation occurs when, while learning an application activity, the learner can simultaneously learn the tool activity for doing the application; or while learning a tool, the learner can simultaneously learn application activities for the tool.

A method is characterized by the statement: "method-act on method-entity to observe (measure) an event". This type of propagation is not usually necessary for the event to occur. That is, the process events may occur whether or not the method activity occurs. However, in the P.E.A. relationship the act is a trigger for the event to occur, or the process explains why the act causes the consequence to occur. A method is an activity for observing or measuring a process, but it is not necessarily a trigger which causes the process to occur, nor is it necessarily an activity which is explained by the process. The relationship is more arbitrary. That is the event may occur whether or not the method activity occurs. Method-process propagation occurs when, while learning a process, the learner can simultaneously learn a method activity for studying or observing the process; or while learning a method activity, the learner can simultaneously learn the process for which the method was devised.

We could describe tool-propagation and method-propagation as two different transactions. They have much in common, but they also have some significant differences. In tool propagation each step of the application is accomplished by one or more steps of the tool activity. There is a direct mapping of the tool activity onto the application activity. In method-propagation the method activity is set up and may be completely independent of the process. After the method is set up the process is triggered. In some cases the activities of the method must also be performed during the execution of the event. In other cases the activities of the method are completed and then the process occurs, then additional activities of the method are completed after the process is complete.

Knowledge .

A propagate transaction requires two or more associated frames. The most common types of propagation are tool-application propagation, in which an application activity is linked to a tool activity; and method-process propagation, in which a process activity is linked to a method activity. In fact, the linking can be considered as between P.E.A. nets with an application P.E.A. net consisting of an application activity, its associated entities, and associated process linked to a tool P.E.A. net consisting of a tool activity, its associated entities, and associated process. A similar linking occurs between a method P.E.A. net and a process P.E.A. net.

Three representations -- structure, function, and physical -- may be required for each of the linked frames in either a tool-application or method-process propagation association. The functional and physical representations usually require dynamic visualization, manipulation, or simulation of the entities involved both as tools and applications, or methods and processes.

Interactions

The primary purpose of the tool-application propagate transaction is to help the learner acquire one set of activities nested in another set of activities. The propagate transaction accomplishes this for the tool-application link by directing the execute transaction to do its work first with one set of activities, the application, and then with the nested other set of activities, the tool. The propagate transaction helps the learner acquire not only the individual skills involved in the activities, but also their close nested relationship.

The propagate transaction chooses an instance of the application activity, directs the execute transaction to teach the learner the steps of the application instance. Then the propagate transaction selects the corresponding tool steps, those required to carry out the steps of the application, and directs the execute transaction to teach the learner these steps of the tool instance. This process is repeated until the entire application has been acquired. Note that a tool-application propagation transaction may teach two or more applications for a given tool set, or it may teach two or more tool sets, or tool step alternatives from a given tool set, for a given application.

An information level interaction presents the steps of the application and illustrates how they are accomplished via the steps of the tool. An information level interaction will usually show only the higher level of the steps in both the application and the tool rather than all the detailed steps. It will not show alternative tool activities for a given application step, or alternative applications for a given set of tool steps.

A demonstration level interaction directs the execute transaction to teach the steps of the application. It then directs the execute transaction to demonstrate how each of these steps is accomplished by completing steps from the tool set.

A manipulation level interaction directs the execute transaction to teach the steps of the application to the learner. The learner is asked to denote (recognize or recall) these steps. The execute transaction is then directed to teach the steps of the corresponding tool. The learner is then directed to actually carry out the steps of the tool to accomplish the step of the application. The propagation transaction manages the interchange between the associated activities and the work of the execute transaction in teaching these related activities.

The primary purpose of the method-process propagate transaction is to help the learner acquire a method for observing or measuring the events of a process. The propagate transaction accomplishes this for the method process link by directing the execute transaction to do its work first with the method activities. The acts performed by the learner as part of the method activities then enable the learner to observe the events of the process which are demonstrated by the interpret shell. The propagate transaction helps the learner acquire the steps of the method and the events which the learner is able to observe or measure because of the execution of these steps.

The propagate transaction chooses an instance of the method activity and directs the execute transaction to teach the learner the steps of the method. Then the propagate transaction selects the corresponding process events, those which can be observed by or measured by the method, and directs the interpret transaction to teach the learner these events. This process is repeated until the entire method and corresponding process has been acquired. Note that a method-process propagation transaction may teach two or more methods, or method alternatives, for a given process, or it may teach two or more processes which can be observed or measured with a given method.

An information level interaction presents the steps of the method and illustrates the events which can be observed or measured via these method steps. An information level interaction will usually show only the higher level of the steps in the method and higher level events in the process. It will not show alternative method activities for a given process event, or alternative processes which can be observed or measured with a given set of method steps.

A demonstration level interaction directs the execute transaction to teach the steps of the method. It then directs the interpret transaction to demonstrate the process which can be observed or measured by theses method steps.

A manipulation level interaction directs the execute transaction to teach the steps of the method activity to the learner. The learner is asked to execute these steps making appropriate observations or measurements of the events of the process. The propagation transaction manages the interchange between the execute and interpret shell as the method is taught and the corresponding events are observed or measured.

Associated transactions

A propagate transaction family involves successive applications of an execute transaction to application and tool activities; or the successive application of execute and interpret transactions to the method activities and corresponding processes. The propagation can be taught at the instance or the class level. At the class level a propagation family may involve the transfer transaction. If observations or measurements require classification, then a classification transaction may be involved; if judgement, then a judgement transaction may be involved.

Analogize[9]

Analogize is the process of learning about one event or process by likening in to a completely different event or process, but one in which some of its events, steps, or their sequence are similar. Analogize transactions involve similarities between functions of the two associated knowledge structures rather than appearance similarities. Analogize transactions usually involve between-domain comparisons rather than within-domain

[9] The authors are indebted to Michael Rosteck, a PhD candidate at Utah State University, for his work on Analogize Transactions for some of the ideas in this section. We take full responsibility for our interpretation of his ideas.

comparisons. Analogies in instruction are far more common than most people are aware. They range from analogous words and phrases, which may involve only a single act or event (e.g. the capacitor forms a *bridge* between ...), to very complex analogies in which two whole complex processes or activities are likened to one another (e.g. a circuit is likened to a system of water pipes and control valves).

Simple analogies, e.g. *bridge* in the previous example, usually require no special instruction except to be sure that the learner has previously acquired the meaning of the analogous terms used. However, complex analogies require a separate transaction shell to enable the learners to make the necessary connections, to avoid overgeneralization, and to form appropriate distinctions between the target content and the analogous content.

Some researchers (Spiro, Feltovich, Coulson, & Anderson, 1989) have suggested that multiple analogies facilitate the understanding of complex phenomena. An adequate analogize transaction should enable the learner to consider the relation of two or more analogies to the target knowledge either simultaneously or sequentially.

Two different approaches could be invoked for the use of analogies in learning. In the first the learner is guided in the selection and use of an analogy by the instruction. In the second, learners are led to form their own analogies. For the present we are concerned only with the first of these options, although a transaction shell could be designed to enhance the learner's ability to form their own.

Capability

An analogize transaction enables the learner to acquire the steps from one activity by likening it to an analogous activity; or to acquire the events in one process by likening it to an analogous process or activity.

Knowledge

An analogize transaction requires two or more knowledge frames, or knowledge P.E.A. nets linked by the relation *analogy for*.

Interactions

The interactions of an analogize transaction must manage two learning processes: (a) the access or retrieval of the analogous knowledge, and (b) mapping the functions of the analogous content onto the target knowledge.

The analogize transaction first directs either an execute or interpret transaction to teach or review the analogous knowledge. It then directs either the execute or interpret transaction to teach all or part of the target knowledge while simultaneously mapping the analogous knowledge to the target knowledge. In the process of the mapping the transaction must make clear where there is correspondence and where there is not correspondence. One of the common problems in using analogies is the danger of "resemblance thinking", that is, the assumption of causality between the two domains being compared. The analogize transaction must enable the learner to see where resemblance thinking could occur and help the learner avoid this false interpretation of the similarities.

An information interaction presents the analogy, the target knowledge, and the principal mapping relationships.

A demonstration interaction presents the analogy, the target knowledge, and the principal mapping relationships, but also includes a far more detailed comparison showing where the analogy holds, where it does not hold, presenting possible instances of resemblance thinking and showing the fallacy of such thinking.

A manipulation interaction presents the analogy, the target knowledge, and then allows the learner to make predictions in a process or to execute an activity based on this analogous knowledge. The feedback system then helps the learner realize where the analogous relationship ends, and where instances of resemblance thinking may have been inappropriately applied. An advanced instantiation of an analogize transaction shell may present the target knowledge and help the learner form their own analogies in an attempt to interpret the target knowledge or execute the target activities.

Associated transactions

An analogize transaction requires either an execute or interpret transaction, or both in order for it to carry out its responsibilities. When class processes or activities are involved various abstraction transactions -- judge, classify, generalize, or decide -- may also be involved.

Substitute

Substitution is the process of learning about one event or process by using what the learner already knows about a similar event or process. Substitution transactions involved knowledge which not only has similar functions, but usually also has similar appearance, The knowledge involved is usually from the same domain. Much of our learning is not completely novel but does in fact involve learning a new activity or process which is only slightly different from a previously learned activity or process. A substitute transaction provides instruction deliberately structured to take advantage of what the learner already knows.

Capability.

A substitute transaction enables the learner to learn an alternative activity or process by comparison, elaboration, or extension of a previously learned activity or process. It also enables the learner to acquire alternative ways to accomplish a given activity or to explain a given process.

Knowledge

A substitute transaction requires two or more knowledge frames linked by the relation *alternative for.*

Interactions

The interactions of a substitute transaction must manage three learning processes: (a) the retrieval of the previously learned knowledge, and (b) mapping the steps or events of the previously learned knowledge to the target knowledge while reviewing the learner's ability to use the previously learned knowledge in the target knowledge, and (c) teaching the learner target knowledge which has no counterpart in the previously learned knowledge.

The substitute transaction first directs either an execute or interpret transaction to review and assess the previously learned knowledge. It then maps corresponding parts of the previously learned knowledge to the target knowledge and assesses the learners ability to use the previously learned knowledge in the target knowledge. It then directs either or both an execute and/or interpret transaction to teach those events and/or steps of the target knowledge that were part of the previously learned knowledge or which were not previously learned.

An information interaction compares the previously learned knowledge with the target knowledge pointing out those aspects which are the same and those parts of the target knowledge which are new.

A demonstration interaction demonstrates the execution or interpretation of those parts of target knowledge that are new.

A manipulation interaction assesses the learners acquisition of the relevant previously learned knowledge, allows the learner to use this previously acquired knowledge with the corresponding steps or events of the target knowledge, and allows the learner to execute the steps or interpret the events of the target knowledge that are new.

Associated transactions

A substitute transaction requires either or both an execute and/or interpret transaction. When class process or activities are involved various abstraction transactions -- judge, classify, generalize, or decide -- may also be involved.

Design

Design is more than a single transaction class. Eventually we anticipate that a number of different types of design transactions will be identified and specified. The designation of a design class here is in anticipation of these future specifications. Hence, we have identified the general capability of a design transaction and the knowledge required but have not specified the nature of the interactions.

Capability

A design transaction enables the learner to use given knowledge frames to *invent* a new activity or entity not previously included in the knowledge structure (EFN). Design includes the creation of new artifacts and is meant to include activities such as *create,* as for a painting or a story; *invent*, as for a new machine or computer software. Design transactions enable the learner to expand the knowledge structure by adding new knowledge frames as a result of creative activity.

Knowledge

A design transaction will usually involve a considerable number of associated frames. The focus of the design activity is to add at least one new entity or activity frame, or to form a new P.E.A. net, not previously included in the knowledge structure. The design activity will require the learner to propagate knowledge from associated frames to the new frame(s).

Interactions

We do not yet clearly understand the interactions necessary for promoting the ability to design and have, therefore, not yet specified these intreactions. We are confident that such an instructional transaction can be specified but leave this specification for the future.

Associated transactions

A design transaction will need to be able to call on all other transactions to assist with its responsibilities.

Discover

Discover is more than a single transaction class. Eventually we anticipate that a number of different types of discover transactions will be identified and specified. The designation of a discover class here is in anticipation of these future specifications. Hence, we have identified the general capability of a discover transaction and the knowledge required but have not specified the nature of the interactions.

Capability

A discover transaction enables the learner to use given frames to *find* a new process not previously included in the EFN. Given a method activity the learner creates new instances of the application of this method and for each instance identifies the events and the causal transformations which explain these events. The learner must then identify an *abstraction model* and class process frame for the instances. This class process then becomes a new principle, model, or theory which is able to explain other instances which are yet to be specified for the knowledge structure. Discover transactions enable the learner to expand the knowledge structure by adding new process frames as the result of discovery or new knowledge.

Knowledge

A discover transaction will usually involve a considerable number of associated frames. The learner will need to acquire the knowledge of related processes and their associated method activities. The learner will use this method knowledge to observe new relationships which form new process frames. The focus of the discover activity is to add at least one new process frame, or to add a process frame not previously included in the knowledge structure, to a P.E.A. net. The discover activity will require the learner to propagate knowledge from associated frames to the new frame(s).

Interactions

We do not yet clearly understand the interactions necessary for promoting the ability to discover and have therefore not yet specified these interactions. We are confident that such an instructional transaction can be specified but leave this specification for the future.

Associated transactions

A discover transaction will need to be able to call on all other transactions to assist with its responsibilities.

References

Gagné; R. M. (1985) *The Conditions of Learning.* New York: 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. (1983) Chapter 9 Component Display Theory. In C.M. Reigeluth (Ed). *Instructional Design Theories and Models: An Overview of their current status.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Merrill, M. D. (1987) Chapter 7 A lesson based on Component Display Theory. In C. M. Reigeluth (Ed). *Instructional Design Theories in Action.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Merrill, M. D. (1988) Chapter 3 Applying Component Display Theory to the design of courseware. In D. H. Jonassen (Ed) *Instructional Designs for Microcomputer Courseware.* Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Merrill, M. D., Li, Z. & Jones, M. K. (in press). Instructional transaction shells: responsibilities, methods, and parameters. *Educational Technology.*

Spiro, R. J., Feltovich, P.J., Coulson, R. L., & Anderson, D. K. (1989). Multiple analogies for complex concepts: antidotes for analogy-induced misconceptions in advanced knowledge acquisition. In S. Vosniadou & A. Ortony (Eds), *Similarity and Analogical Reasoning*. Cambridge: Cambridge University Press.