Instructional Transaction Theory:

Knowledge Relationships among Processes, Entities, and Activities

M. David Merrill and ID_2 Research Team[1]

Utah State University

Revised November 1992

Abstract

In previous papers the authors have introduced Instructional Transaction Theory (See Merrill, Li, and Jones, 1990a, 1990b; Jones, Li, and Merrill, 1990; Merrill, Li, and Jones, 1991, 1992; Merrill, Jones, and Li, 1992). This paper is a further elaboration of some of the ideas previously presented.

The representation of knowledge in an electronic knowledge base is a necessary component of an automated instructional design expert system. We have previously suggested a syntax for such knowledge representation. This syntax consists of process, entity and activity knowledge frames that are elaborated by way of components, abstraction, and associations. In this paper we identify knowledge properties as that integrating component of knowledge that defines an important type of association among knowledge elements. We further introduce the idea of a PEA-Net, a knowledge structure consisting of processes, entities and activities related in such a way as to provide an integrated whole.

A knowledge base composed of PEA-Net structures makes it possible to automate a variety of instructional interactions that enable the learner to acquire this knowledge. Attaching these PEA-Net knowledge structures to appropriate media resources enables the knowledge base to support experiential simulations. Experiential simulations enable the learner to acquire a mental model of the phenomena being taught.

Knowledge Representation

<u>The Importance of Knowledge Representation</u>

Instructional Transaction Theory attempts to provide a sufficiently detailed prescriptive instructional design theory to enable the development of an automated instructional design expert system (ID Expert). We based Instructional Transaction Theory on the Gagné assumption that there are different kinds of instructional outcomes

---

[1] Instructional Transaction Theory is a product of the ID_2 research program in the Department of Instructional Technology at Utah State University. This work is a team effort with significant contributions from all members of the team. The author specifically acknowledges the contributions of his colleagues Zhongmin Li, Mark K. Jones, Jennifer Chen-Troester and Scott Schwab to the contents of this paper.. The author also acknowledges the many suggestions from graduate students who have struggled with these ideas in the author's instructional design classes.

or goals, and that each kind of outcome requires different conditions of instruction (learning strategies). Instructional Transaction Theory enables the design of computer-based instructional algorithms (instructional transaction shells) that will meet the required conditions for each of these different instructional outcomes. A given transaction shell can teach different learning tasks within a given knowledge domain, or teach learning tasks among different knowledge domains.

A critical element of Instructional Transaction Theory is knowledge representation. Without a formal and consistent syntax for this knowledge representation it is not possible to build instructional algorithms that teach this knowledge. If the subject matter for different topics and different domains use the same syntax, then the same instructional algorithm (transaction shell) can teach different topics within a given subject matter domain, or subject matter from different subject matter domains. We assume that instructional strategy (as represented in an instructional transaction shell) is somewhat independent of the knowledge taught. We assume that the same strategy can teach different topics and even different subject matters.

For purposes of developing an instructional design expert system we assume that a knowledge base external to the learner will represent the knowledge (the information and skills to be learned). We make no claims about how a learner organizes and elaborates their cognitive structure, since Cognitive Scientists do not agree on these matters. We stand on the weaker, and more defensible assumption, that we can analyze the organization and elaboration of knowledge outside the mind, and presume that there is some correspondence between these and the representations in the mind.

Gagné and Merrill (1990) suggested the idea of an *enterprise.* An enterprise is a complex human performance involving an integrated set of knowledge and skill. While different conditions (learning algorithms) are necessary for the learner to acquire each of the component knowledge and skill elements of the enterprise, it is necessary for the learner to integrate this knowledge into a consistent whole or mental model. We assume that the various knowledge components are related in ways that facilitate the acquisition of the integrated knowledge required by a learned enterprise.

In this paper we suggest that the association between a process, entity and activity (a PEA-Net) comprises one very important way to represent integrated knowledge. We define the knowledge *property* as a shared knowledge component. The links between knowledge components are the effects that one knowledge component has on the properties of an associated knowledge component.

Knowledge Representation in Instructional Transaction Theory

We proposed an Elaborated Frame Network for this formal knowledge representation syntax (Merrill, Li, and Jones, 1990b; Jones, Li, and Merrill, 1990). We propose to represent knowledge with objects that we call *frames*; each frame has an internal structure (slots, which contain values for the structure), and links to other frames. These slots (both internal and external) are *elaborations* of the frame. The set of all elaborated frames together, which contains all the knowledge to be instructed for a given human enterprise (a set of interrelated activities and processes), is called an *elaborated frame network (EFN)*.

We propose that there are three fundamental knowledge frame types: *processes*, that are sets of related events that occur in the natural world either independent of the learner or as a consequence of some action by the learner; *entities*, that correspond to something such as a device, object, person, creature, place, or symbol; and *activities*, that are sets of related actions to be performed by the learner.

Frames in an Elaborated Frame Network have three types of elaboration: *components*, that correspond to the events (which are also process frames) of a process, to the parts (which are also entity frames) of the entity, and to the steps (which are also activity frames) of an activity; *abstractions*, that correspond to a class-subclass-instance hierarchy into which the frame is classified; and *associations*, that are meaningful links between process, entity, and activity frames in the network.

The network structure of the knowledge representation allows information to move through the structure, so that data contained in one part of the net affects the data stored elsewhere. Two principal means by which this occurs are: *inheritance*, by which properties and components of a class or superclass in an abstraction hierarchy are passed to a subclass or instance; and *propagation*, by which the knowledge of one frame changes the knowledge of another frame connected to it through an association link.

The purpose of this paper is to describe more fully the propagation of knowledge in that portion of the Elaborated Frame Network represented by linked process, entity, and activity frames -- a *PEA-net*. A PEA-net is an association that frequently occurs in most subject matter. The entity is that object, or those objects, which are acted on; the activity is that which is done by the learner either with or to the entity; and the process is the consequence of this activity in terms of some event or set of events that change the properties of the object in some way. Learning about the entity helps the learner understand what the object is in terms of its parts, their location, and their function. Learning the activity helps the learner acquire the skills for manipulating the object. Learning the process enables the learner to know the consequence of their acts and to make predictions about future consequences.

A fundamental assumption of this paper is that association links are not idiosyncratic. Certain associations are necessary to build adequate knowledge representations. In other words, whenever there is an activity or process in a knowledge base, there is always a potential link to its PEA-net partners. There is always an associated entity affected by an activity and an associated process that is the consequence of the activity. Whenever there is a process in a knowledge base there is always an associated entity whose properties are changed by the transformation of the process and often an associated activity on which the occurrence of the process is conditional.

Properties

In previous papers we have identified properties as a fourth elaboration of an Elaborated Frame Network and we ambiguously defined properties and their role in knowledge propagation. In this paper we will more adequately define properties and identify their role in knowledge propagation within a PEA-net knowledge structure.

A property is a qualifier or quantifier associated with a process, entity, or activity or a component of a process, entity, or activity. A property has a set of legal values and a particular value in a given situation. *Property values* are abstractions such as numbers, relationships (above, below, inside), qualities (color), and kind (male, female).

Properties may be shown as separate frames but are only meaningful when linked to a process, entity, or activity. A process, entity, or activity to which a given property is linked is called the *property owner.*

Properties may appear in their own abstraction hierarchies -- class-subclass structures. However, properties are always classes and are usually high (more abstract and less concrete) in their respective abstraction hierarchies. A property by itself cannot be an instance. It is only an instance when it is a property of some other instance frame -- a process, entity, or activity instance. An EFN for a given enterprise will usually not include the abstraction hierarchy associated with the property unless the property is the focus of the instruction.

A subclass or instance in an abstraction hierarchy inherits the components, the properties, and the properties linked to the components, of its superclass frame. An inherited property inherits all the legal values but may not inherit the specific value associated with its parent.

Properties propagate across all association links. That is, the properties, whose values are defined by an activity, are the properties of an entity associated with this activity. The properties, whose values are changed by a process, are the properties of an entity associated with this process. The conditions necessary for a process to occur are values of properties of a process, entity, and or an activity associated with the process. Propagation across association links means that associated knowledge frames are both elaborated with the same properties.

PEA-Net Propagation

Specific associations are required to adequately define the knowledge within and among knowledge frames. A PEA-net knowledge structure implies certain required association relationships among process, entity, and activity knowledge frames. In previous papers descriptions of an EFN allowed unspecified associations (meaning that we allowed the user to specify the relations involved). We previously suggested the following relations: ENTITY ----uses/used by ----- ACTIVITY; ENTITY ---- required/required by ---- PROCESS; and ACTIVITY ---- involves/involved by ---- PROCESS.

These relations are too ambiguous. In this paper we replace them with the following:

In a PEA-net a PROCESS has four primary association relations: *located-in* an entity; *conditional-on* properties of an entity, activity or process; *changes-the-properties-of* an entity; and *condition-for* a process or activity. *Located-in* identifies an associated entity or part of an entity within which, or upon which, the transformation occurs. *Conditional-on* identifies an associated process, activity, and/or entity that has one or more properties whose values constrain the performance of the process. *Changes-the-properties-of* identifies one or more entities whose property values are changed by the transformation of the process. *Condition-for* identifies one or more associated processes whose performance is conditional on one or more property values associated with the performance of the transformation or one or more of the entity property values changed by the transformation.

In a PEA-net an ENTITY has three primary association relations: *changed-by* a process, *acted-on-by* an activity, and *location-of* a process. *Changed-by* identifies an associated process that changes the values of one or more properties of the entity. *Acted-on-by* identifies an associated activity that defines one or more property values of the entity. *Location-of* identifies an associated process that occurs within, or upon, the entity or one or more of its components.

In a PEA-net an ACTIVITY has three primary association relations: *conditional-on* properties of an entity, activity, or process; *acts-on* an entity; and *condition-for* a process or activity. *Conditional-on* identifies an associated process, activity, and/or entity that has one or more properties whose values constrain the execution of the activity. *Acts-on* identifies an associated entity that has one or more properties whose values are defined by the activity. *Condition-for* identifies an associated process or activity whose performance or execution is conditional on one or more activity property values associated with the execution of this activity or one or more entity properties defined by this activity.

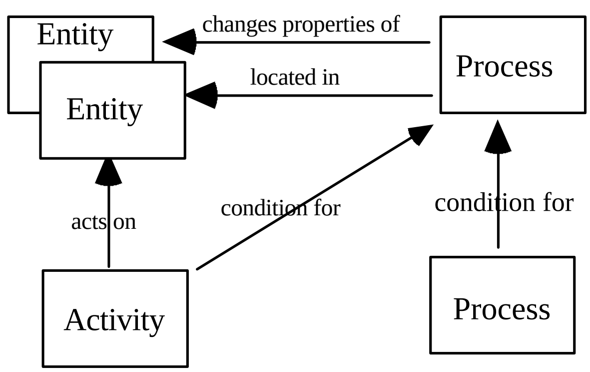

Figure 1 Primary association relations in a PEA-net knowledge structure

Figure 1 shows a configuration of processes, entities, and an activity. The association links are shown by arrows and the nature of the propagation represented by this link is indicated. Thus the process in the upper right is conditional on the process in the lower right; the process in the upper right is located in one entity and changes the properties of another entity. Both of these entities may be parts of a larger entity. The activity acts on one entity and is simultaneously a condition for the process.

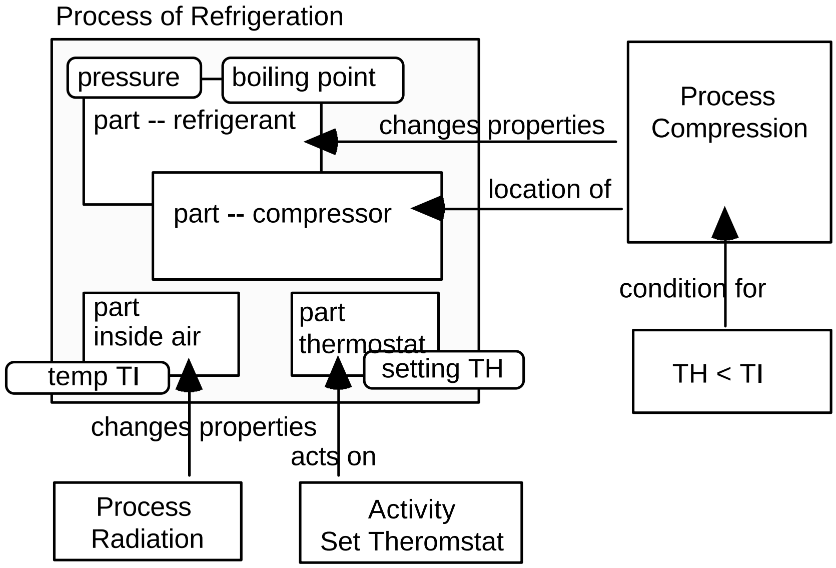

Figure 2 Partial PEA-net knowledge structure for refrigeration

Figure 2 shows a configuration of processes, entities (parts), and an activity represented by our refrigeration example[2]. The process of compression is conditional on the relationship of the setting of the thermostat (TH) and the temperature of the inside air (TI). This process is located in the part compressor and changes the value of the properties, pressure and boiling point, of the part refrigerant. The process of radiation changes the value of the property, temperature, of the part inside air. The activity of setting the thermostat changes the value of the property, setting, of the part thermostat.

The Knowledge Base

The knowledge base is the formal representation of the knowledge to be taught. In a given implementation this knowledge base can take many different forms. For convenience in this paper we have used a tabular representation for the various slots in a knowledge frame. Table 1 is a knowledge base for a process frame (refrigeration) which consists of 6 events or sub processes. The rows associated with each event name represent a sub process or event. Table 2 represents an associated process (radiation) the consequences of which are conditions for some of the events represented in Table 1. Table 3 represents several associated activity frames that also define property values on which the events of Table 1 are conditional. Table 4 is the knowledge base for the entity frame (the refrigerator) and its subparts that serve as the location for the events of Table 1 and as the entity that is changed by the transformations that occur when the events of Table 1 are enacted.

---

[2] The authors thank David Macaulay (1988) for inspiration concerning refrigeration. Nevertheless, any errors in our representation of this process should be attributed to the authors and not to Mr. Macaulay.

Refrigeration	cyclical					
Event Name	Location	Condition	Transformation	Property	Owner	To value
compression	compressor	TH < TI	increases	pressure	refrigerant	high
			increases	boiling point	refrigerant	10
condensation	condensor	BP > TR	changes	form	refrigerant	liquid
			increases	temperature	refrigerant	9
radiation	outside air	TO < TR				
		E=not enclosed	decreases	temperature	refrigerant	TR = TR -1
		E = enclosed	decreases	temperature	refrigerant	TR = TR - 1
			increases	temperature	outside air	TO = TO + 1
expansion	expansion valve		decreases	pressure	refrigerant	low
			increases	boiling point	refrigerant	0
evaporation	evaporator	BP < TR	changes	form	refrigerant	gas
			decreases	temperature	refrigerant	2
radiation	inside air	TI > TR	decreases	temperature	inside air	TI = TI -1

Table 1. Knowledge base for the process refrigeration

Detailed Description of the Process Knowledge Base

The knowledge base for each event in a process consists of the following slots: the event name, the location where the event occurs, the conditions that must be met in order for an event to occur, the transformation that comprises the event, the property that is changed as a result of the transformation, the owner of the property, and the value that the property acquires as a result of the transformation. In the following paragraphs we describe each of these elements of the process knowledge base.

The order of slots in Table 1 facilitate reading the table and make it easier for a subject matter expert to describe the process using the conventions of the knowledge base. For example, read the table in the following manner:

The event *compression*, located in the *compressor*, occurs if the *setting of the thermostat* (TH) is less than the *temperature of the inside air* (TI). Compression *increases* the *pressure of the refrigerant* to a value of *high* AND *increases* the *boiling point of the refrigerant* to *10.*

The event *condensation*, located in the *condenser*, occurs if the *boiling point of the refrigerant* (BP) is greater than *the temperature of the refrigerant* (TR). Condensation *changes the form of the refrigerant* to *liquid* AND *increases the temperature of the refrigerant to 9.*

Event name

This is the identifier for the event. Since this name will be used in generated presentations to the student, use descriptive names rather than abbreviations.

Event location

This is the part of an entity in which or on which the event occurs. One of the learning tasks for the learner is to locate the part of the entity in which a given event occurs. Since the name of the location will also be used in generated learner presentations, use descriptive and accurate names rather than abbreviations.

Condition

A condition is an expression that must be evaluated *true* in order for the transformation to occur. State conditional expressions in terms of properties. For convenience use symbols to represent the properties. The properties involved in a condition are owned by entities, activities or processes.

A condition constitutes an IF statement. If the condition is *true*, then the transformation occurs. If the condition is *false* the transformation does not occur and there is no change in the value of the properties. A condition can also be an IF-THEN-ELSE statement. In this case the word ELSE appears in the second row or the condition column and the transformation following the else is enacted when the condition is false. See the radiation event in the associated event of Table 2.

An event may have more than one condition. In the example, the first radiation event has two conditions: *TO < TR* and *E = not enclosed*. The transformation appears on the line following the second condition to indicate that both conditions must be satisfied before the transformation is enacted. The condition, *E = enclosed* indicates the alternative situation when the first condition, *TO < TR,* is satisfied and the second condition, E = enclosed, is true. The transformation following the *E= enclosed* condition is enacted under this condition.

When no condition is indicated, as for the event *expansion*, then the transformation always occurs whenever this event is enacted.

Transformation

The transformation indicates the nature of the change in property values that occurs as a result of the event. There are three primary types of transformations: *move, change, and increase/decrease*. The type of value allowed depends on the nature of the transformation. The value affected by a *move* transformation is a change in location. The value is some location indicator. The value affected by a *change* transformation is a change in form, appearance, or state. The value is some form, appearance, or state indicator. The value affected by an *increase/decrease* transformation is a change in quantity. The value is a number of some type.

Synonyms can be substituted for move, change, increase/decrease. The knowledge base should contain an equivalence table for such synonyms. For example: synonyms for move could be turn, exchange positions, lift. Such an equivalence table is not included in this paper.

A given event may have more than one transformation. The convention used in Table 1 is to list the second transformation on a second line of the table. This indicates that the event results in both transformations. For example, compression increases the pressure which in turn increases the boiling point. These could have been shown as separate events but for simplicity are shown as a single event in this knowledge base.

Property

The property is that which is changed by the event. A property is usually a location, state, or quantity of some property owner. The value assumed by the property is changed by the transformation.

A given transformation could change more than one property. The convention used in our table would be to list the second property on a second line of the table. In the example given we have shown a one-to-one correspondence between properties and transformations.

Property owner

The property owner is the entity that is qualified or quantified by the property.

To value

To value is the value that results from the transformation. This value must be consistent with the type of transformation: locations for move; state, appearance, or form indicators for change; and quantities for increases/decreases. The value can be a constant, string, or an expression. An expression is a precise description of the transformation. Expressions are always in terms of relationships between properties. Symbols may be used for such expressions.

The actual property values are not shown on the process knowledge base but are included in the knowledge base representing the entity that owns the property (see Table 4). Property values are dynamic and are changed as a result of the process. Each property also has a default value that is restored to the value when an enactment of a process is reset.

Other conventions used in Table 1 are as follows: The term cyclical[3] appearing in the first line of the table indicates that during an enactment of the process the first event follows the last event. The order of the events in the table is significant. During an enactment the enactment method[4] evaluates each event in turn. If the condition is true, then the transformation occurs and the resources associated with the true condition are displayed. If the condition is false and no “else” transformation is indicated, then the transformation does not occur and the resource associated with the false condition is displayed. If the condition is false and there is an “else” transformation, then the “else” transformation occurs and the resources associated with this condition are displayed. The enactment method then evaluates the next event. In a cyclical event this enactment continues with each event in turn until the enactment is stopped. While events follow one another in sequence they do not necessarily depend on one another. Only the conditions of a given event control how the event is enacted.

Associated Processes

[3] Cyclical is only one type of process. Other types include sequential and parallel processes.

[4] A method is a piece of computer code which is part of a transaction shell. For a discussion of transaction shell responsibilities, methods and parameters see Merrill, Li, and Jones, 1992.

Table 1 is the knowledge base for the process of refrigeration. However, the process is more interesting and complete if the associated process of radiation of heat from the air into the refrigerator is also included. In this way the refrigerator slowly warms up causing a change in the value of the property TI (temperature of the inside air). This process is also a consequence of the activity of opening the refrigerator door that causes the radiation to occur much more quickly.

Associated Events						
Name	Location	Condition	Transformation	Property	Owner	To value
radiation	inside air	D = open	increases	temperature	inside air	TI = TI + 2
		else	increases	temperature	inside air	TI = TI + .2

Table 2. Knowledge base for process of radiation associated with refrigeration

Associated Activities

Refrigeration can occur, and can be described without any associated activities. However, interpretation becomes more interesting when the learner can affect the functioning of the refrigerator by various activities. We have included only a few, such as opening the door, setting the thermostat, and enclosing the refrigerator in a closed space so that the outside air gets warmer and affects the operation of the refrigeration process. These activities can be performed in any order and at any time during the execution of the refrigeration process. These activities affect values of properties that are conditions for refrigeration and hence modify the enactment of the process.

Associated Activities						
Act Name		Condition	Act	Property	Owner	To value
Open door			move	location	door	open
Close door			move	location	door	closed
enclose			change	environment	refrigerator	enclosed
do not enclose			change	environment	refrigerator	not enclosed
set thermostat			move	setting	thermostat	TH

Table 3. Knowledge base for activities associated with refrigeration

Activities are described using a knowledge base very similar to the process knowledge base. An act directly or indirectly changes the value of a property. Acts, like transformations, can either move an entity, change the state, appearance, or form of an entity, or change the quantity of a property of some entity. We will discuss conditional activities in a subsequent paper.

Associated Entities

A process is defined as a transformation of the values of the properties of some entity. The entities in Table 4 were derived from Tables 1, 2, and 3. Entities appear in the process knowledge base in the column headed *location*, and in the column headed *owner*. The properties of these entities appear in the column headed *property*. The symbols must be assigned by the SME to be consistent with the expressions appearing in the *condition* and *to-value* columns. The default values are assigned by the SME as a starting place for the enactment of the process. The dynamic values are those determined by the transformations of the process or the acts of the activity. These values are reassigned with each enactment of the process or by each execution of an activity. If a transformation does not change a value then the previous value is retained until the process is reset at which time the default values are restored.

Not all entities need properties. In the example, the locations where the events of the process occur -- compressor, condenser, expansion valve, and evaporator -- do not have properties that are affected by either the process of refrigeration or the activities associated with this process. We have not analyzed the entity to indicate the clustering of parts within a single entity, the refrigerator.

Associated Entities				
Name	Properties	Symbol	Dynamic value	Default value
compressor				
condensor				
outside air	temperature	TO		7
expansion value				
evaporator				
inside air	temperature	TI		7
refrigerant	pressure			low
	boiling point	BP		0
	form			gas
	temperature	TR		7
door	location	D		closed
refrigerator	environment	E		not enclosed
thermostat	setting	TH		7

Table 4. Knowledge base for entities associated with refrigeration

Resources

Resources are particular mediated representations of the subject matter to be taught. The knowledge base is generic, not specific to any particular media device, whereas resources are specific pieces of graphic, video, and/or audio. A given knowledge base can be associated with two or more different sets of resources. One can replace a resource without affecting the knowledge base to which it is associated.

When an instructional transaction is enacted, then the methods of the transaction display or compile particular resources for the learner to observe or manipulate as part of the instruction. The actual representation presented to the learner may be a generated resource, one constructed from the knowledge base by the method of the transaction; or a resource from a resource data base associated with a particular knowledge base. In this section we will illustrate some of the resources that may be associated with the knowledge base previously described.

Generated Text Resources

Generated resources are those which are constructed by the method of the transaction from the information provided in the knowledge base rather than being specific text or graphic messages stored in a resource data base.

Diagrams

A transaction method can construct a block diagram or flow chart representing the events in a process or the steps in an activity from the information given in the knowledge base. For example, Figure 2 is a block diagram representing the process of refrigeration. This diagram is generated by the transaction method from the information provided in the knowledge base. The information that the process is cyclical is a parameter that the generation algorithm uses to include a return arrow in the diagram as shown.

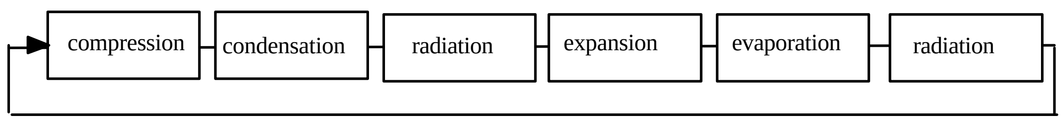

Figure 2 Process block diagram for REFRIGERATION

The transaction can provide a variety of instructional interactions with this generated graphic. The learner can be asked to match the names of the events with the block corresponding to that event. The learner can click on a block to see an explanation of the event (see the next section on Generated Text). The learner can be asked to match an explanation to the appropriate block.

Generated Text

The information included in the knowledge base also enables a transaction method to generate English like statements explaining a given event. The method uses a text template, filling in the slots of the template with information from the knowledge base. Such text statements can then be used to present information to the learner concerning the events or to test the learners' ability to recognize or even recall the events of the process. Similar text templates can be generated for activities.

The following is a text template for generating statements about any event:

IF < conditions > then < event > < transformation> the < property > of the < property owner > to <to value>. Information for the slots < > is obtained from corresponding cells of the knowledge base. The second phrase is repeated for each property that is changed. If there is no condition then the explanation starts with the < event >. Using the symbol table conditions are translated into words by means of the following text template: the < property > of the < property owner >**.** Operators are translated into their English equivalent.

For example the following is text is a generated explanation of the event compression:

"IF the *setting* of the *thermostat* is less than the *temperature* of the *inside air*, then *compression increases* the *pressure* of the *refrigerant* to *high*. *Compression increases* the *boiling point* of the *refrigerant* to *10*."

A similar explanation can be generated for each of the events of the process.

<u>Text Resources</u>

The author can write and store specific text messages in a resource data base. These resources are indexed to the knowledge base.

One such text resource may be associated with the entire process of refrigeration as follows:

> Refrigeration is the transfer of heat from inside an enclosed space to the outside air.
>
> Hot things **radiate** heat rays, which are absorbed by cooler objects.
>
> A **refrigerant** flows through a closed system of pipes. Inside the refrigerator the refrigerant is colder than the inside air and absorbs heat from the inside air. Outside the refrigerator the refrigerant is hotter than the outside air and radiates heat to the outside air. In this way the refrigerant carries heat out of the refrigerator.

Such a text resource would be displayed by the method whenever the parameters are set to display text explaining the whole process. Bold words represent hot words to which additional hypertext resources can be attached.

Text resources can also be written and associated with each of the events in the process. Such resources are than displayed whenever the transaction wants to explain the event with a text resource from the resource data base rather than using generated text explanations. Such a resource for the event *compression* is as follows:

> An electric refrigerator contains a compressor to move a **refrigerant** (a volatile liquid) around a pipe. The compressor pumps the liquids from the evaporator into the condenser. It then returns through the expansion valve. The compressor puts the refrigerant vapor under **high pressure**.

By setting an appropriate parameter a transaction can present author supplied resources rather than generated resources in the interactions with the learner. With appropriate multimedia hardware and software audio resources can be substituted for text resources where more appropriate.

<u>Graphic Resources</u>

There are several levels of graphic displays that can be associated with a knowledge base: a base or process graphic, event graphics, property graphics, and value graphics. A base or process graphic is associated with the entire process.

Figure 3 is a process graphic for refrigeration.

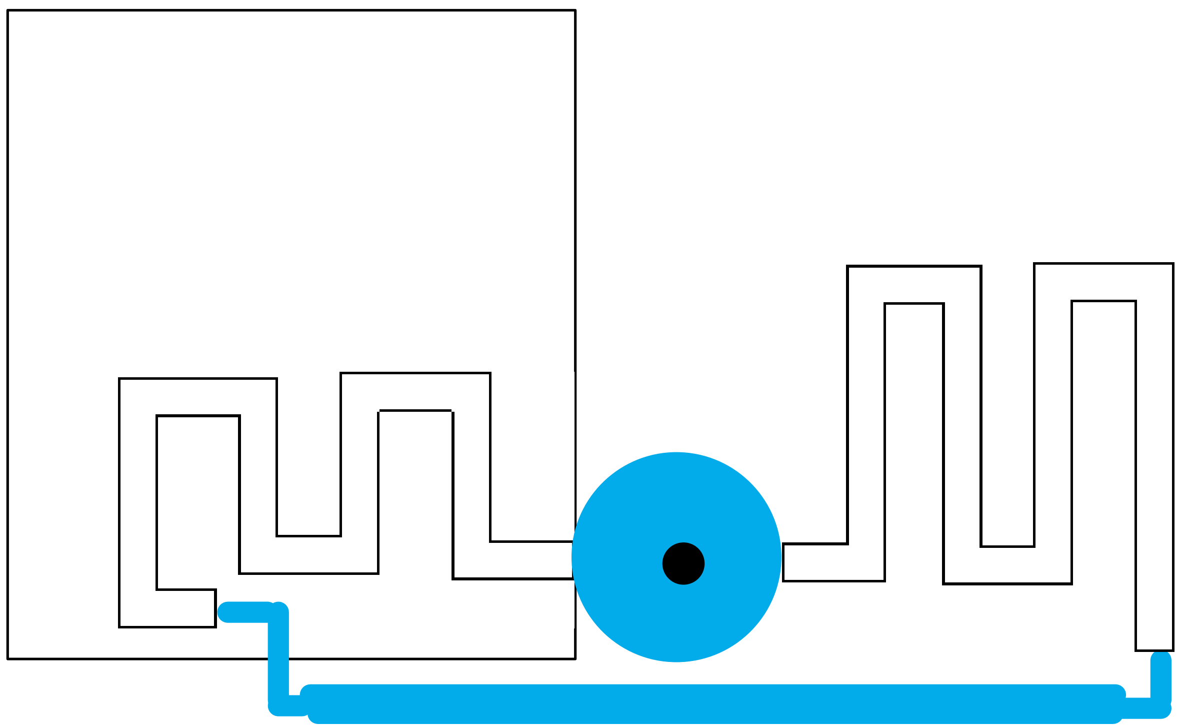

Figure 3 Process resource for refrigeration

An event graphic is associated with a particular event. Event graphics often overlay a process graphic to make a composite representation for the learner. It is not necessary for every event to have an associated graphic.   Event graphics often come in at least two versions: the appearance when the event is enacted and the appearance when the event is not enacted. In some cases a single graphic is displayed when the event occurs and no graphic is displayed when the event does not occur (See Figure 6).   Figure 4 is an event graphic associated with “condensation”. The left figure is the appearance when condensation does not occur. The right figure is the appearance when the condensation does occur. Figure 5 is an event graphic associated with “evaporation”. The left is the appearance when evaporation does not occur; the right is the appearance when evaporation does occur.

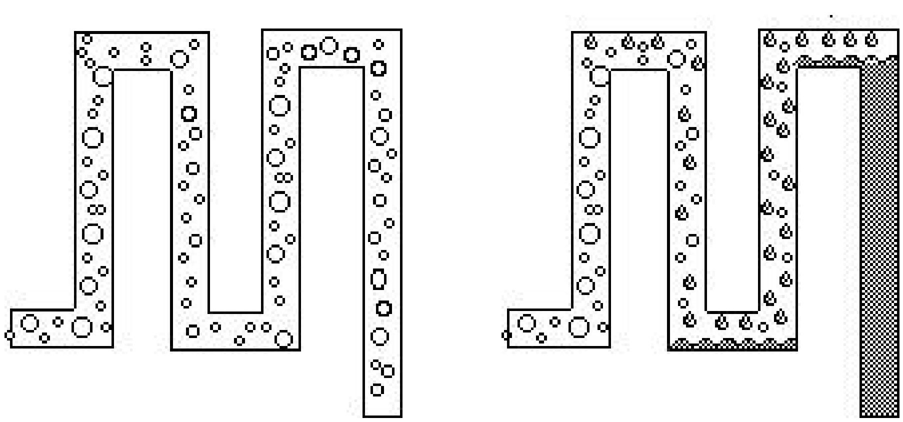

Figure 4 Event graphic resources for condensation

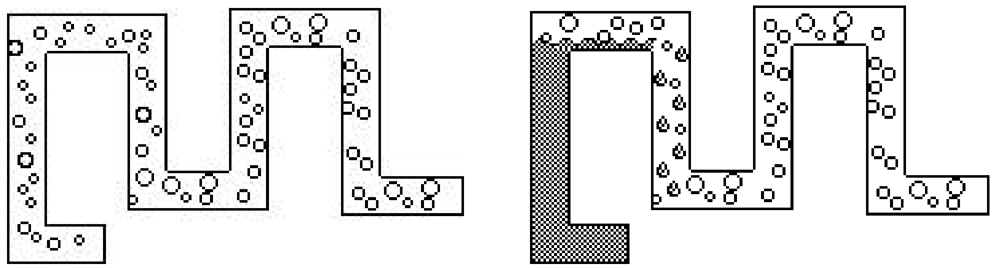

Figure 5 Event graphic resources for evaporation

Figure 6 is another event graphic. The left graphic is shown when radiation occurs inside the refrigerator and is not shown when radiation does not occur. The right graphic is shown when radiation occurs outside the refrigerator and is not shown when radiation does not occur.

Event graphics could also consist of audio or video resources that are displayed when a given event occurs.

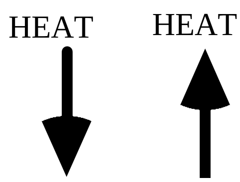

Figure 6. Event graphic resource for radiation

Property graphics are associated with a particular property. Property graphics often overlay process and event graphics to provide a composite representation for the student. It is not necessary for every property to have an associated graphic. Property graphics are usually objects that contain their own methods of how to display their various value states. In this case the property value is supplied to the property graphic and it displays itself with the appropriate state. Figure 7 is a property graphic for the property setting of the thermostat. The dark rectangle is the value indicator. The learner sets the thermostat (an activity) by dragging the indicator to the appropriate value.

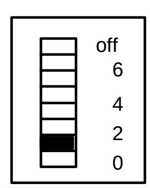

Figure 7 Property graphic for setting of thermostat

Property graphics can be either display only or display with input. If the graphic is an input display then the learner can set the value by moving a value graphic to a new location or by entering a value as in Figure 7,

Figure 8 is a property graphic for the property temperature. This same graphic is used for temperature of the refrigerant, temperature of the inside air, and temperature of the outside air. The location of the graphic is different according to the value indicated. The mercury bar is the value graphic with a different length mercury bar for each of the values that the temperature can assume.

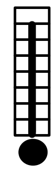

Figure 8 Property graphic for temperature of the refrigerant, inside and outside air

Figure 9 is another property graphic for the pressure of the refrigerant. The needle is the value graphic with its position changing for the values low and high.

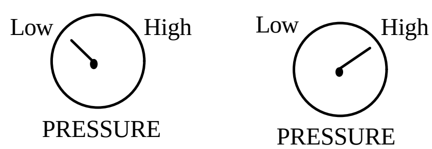

Figure 9 Property graphic (GP) for pressure of the refrigerant

Instructional Transactions

Simulation based on the Knowledge Base

Figure 10 shows the process graphic overlaid with the event graphic for condensation, for radiation to the outside air, for evaporation, and for radiation from the inside air. The property graphic of the thermostat and temperature of the inside air is also shown. A small representation of the whole refrigerator is also represented with the door open. This graphic is an act graphic for the act of opening the door. This is a simulation of the refrigerator that enables the student to interact with the processes.

The learner can engage in the activity to set the thermostat by dragging the indicator to the desired setting. The learner can engage in the activity to open the door by clicking on the refrigerator graphic. As long as the mouse button is down the door is open, when the mouse button is released the door closes. (The door is shown in the open position.) When the door is open the radiation event shown in table 2 executes. Each execution of this event increases the temperature of the inside air a small amount. Hence, as the door is held open, the temperature of the inside air, as shown by the thermometer, increases.

As soon as the temperature of the inside air exceeds that of the thermostat, then compression begins. There is no graphic associated with compression so the representation for the student does not change. However, the pressure of the refrigerant is changed to high and the boiling point of the refrigerant is changed to 10 as indicated by the knowledge in table 1. These values cause the event of condensation to occur. This causes the graphic associated with condensation to be displayed as shown in figure 10. Radiation also occurs causing the event graphic from radiation to the outside air to be shown. And so forth. Each event in the process is enacted causing the graphic for evaporation to occur and the graphic for radiation from the inside air to occur.

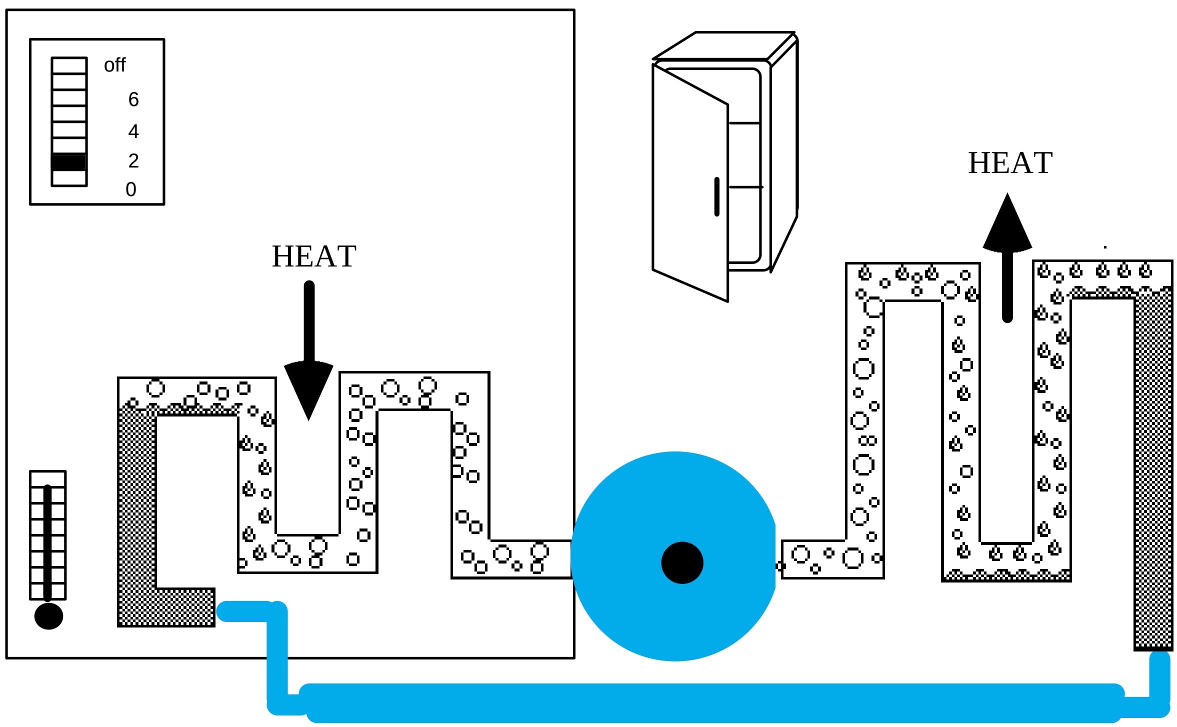

Figure 10 Experiential simulation of the process of refrigeration

When the temperature of the inside air is equal to the temperature of the thermostat, then compression stops. This in turn changes the appearance of condenser, removes the radiation graphic, changes the appearance of evaporation, and removes the inside air radiation graphic.

This process continues to execute as long as the simulation is on.

The transaction could also be modified to show the process of refrigeration one event at a time. The learner could click on the compressor and see the property values of the refrigerant at the point after compression and before condensation by means of a value table or by the temperature and pressure graphics. The learner could also receive an explanation of the event via the generated text explanation. Enactment is suspended at this point. The learner can then examine the next event by clicking on the condenser. The graphic, as shown in figure 10 shows the change in form of the refrigerant from a gas to a liquid. The learner could also be shown the temperature of the refrigerant and the boiling point of the refrigerant. This demonstration could be continue until the entire process had been demonstrated at which the simulation could be resumed to allow the student to observe the integrated process as a whole.

Conclusion

In conclusion we would like to make the instructional design implications of the points made in this paper more obvious.

We suggested that an appropriate knowledge representation would enable the automatic development of interactive experiential instruction through appropriate instructional transaction shells. We also suggested that one form of integrated knowledge consists of PEA-Net associations consisting of associated processes, entities, and activities that share common properties that are modified by the processes and activities involved.

It is proposed that many different processes from a variety of knowledge domains can be represented by the PEA-Net knowledge representation suggested here. Authoring consists of describing the process in terms of the associated process and sub events, the entities and parts affected by the process, the activities associated with the process, and the properties shared by these processes, entities and activities. The authoring process must also include the selection or creation of resource objects that correspond to the process, events, and properties. If the described syntax for knowledge representation is followed, then the resulting knowledge base can be enacted to produce descriptions, demonstrations, and experiential simulations of the phenomena being taught.

Representing knowledge with PEA-Net associations and properties enables the construction of transaction shells that know how to teach this knowledge in a variety of ways with no further authoring required. Once the knowledge has been specified and appropriate resources have been linked to this knowledge, the development of instruction is automatic. This instruction can include tutorials that drill the learner in the names of the events, the involved parts, the order of the events, and the recognition of the appearance of the entities when a given event occurs. The instruction can also include experiential simulations of the event in which the student can perform appropriate actions and observe the consequence of these actions in the enactment of the process by way of the resources. Both the tutorials and the experiential simulations can consist of presentations to the student or assessment of the learners' ability to recall or recognize the parts, events, and activities and the learners' ability to identify errors and make predictions concerning the events in the process. The transactions also enable the learner to confirm these predictions by experiential simulations of the phenomena being taught.

References

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

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

Macaulay, D. (1988). *The Way Things Work.* Boston: Houghton Mifflin Company, .

Merrill, M. D., Li, Z., & Jones, M. K. (1990). Limitations of first generation instructional design (ID_1). *Educational Technology*, 30(1), 7-11.

Merrill, M. D., Li, Z., & Jones, M. K. (1990). Second generation instructional design (ID_2). *Educational Technology,* 30(2), 7-14.

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

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

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