Modeling Instructional Content

James A. Hadley
Consultant
jhadley1@gmail.com

ABSTRACT

In order to help learners acquire strong knowledge structures or mental models of performance and systems, they must be exposed to instructional messages and environments which convey the complexity of the models they are intended to assimilate. Many instructional strategies and media focus on helping learners build and develop these mental models. However, before strategies and media can be selected, instructional designers must analyze content into structures which are similar to the mental models the learner will build.

Key to analyzing and building these structures is identifying the relationships between systems, environments and performance. The common structures for defining these relationships are task and learning objective statements. These structures identify a performance requirement for a system within a given environment. These statements, by their very nature though, are static and discrete and do not encompass all the dynamic relationships which exist in real-world operation.

Modeling approaches can be used to help instructional designers develop strong and complete knowledge structures of content. The very essence of modeling helps identify key relationships between systems and the environments they exist within. Unfortunately, modeling methods are not commonly associated with traditional instructional systems design. Rather, modeling skill sets are more associated with systems analysis, intelligent tutoring and computer science. This paper identifies and introduces modeling methods for instructional designers to capture the relevant performance, systems, and environmental knowledge for representations in training.

ABOUT THE AUTHOR

James A. Hadley is an instructional design consultant focusing on advanced instructional approaches and technologies. He has a Masters degree in Instructional Technology from Utah State University. James has spent the last five years developing instructional simulations for the Navy Integrated Learning Environment and for commercial clients such as Pfizer. His research interests focus on the interplay between instructional analysis and design and identifying modeling and simulation methods for instructional designers.

Modeling Instructional Content

James A. Hadley
Consultant
jhadley1@gmail.com

INTRODUCTION

One of my favorite activities during a seminar to instructional systems design students is to have two groups come to the whiteboard and complete a task analysis. The first group, is told to analyze and write learning objectives for fixing a flat bicycle tire. This group lists the necessary steps and tools required to do the job. These tasks easily translate into standard learning objective statements.

The second group is also told to write down a task analysis and set of learning objectives, but instead of fixing a flat bicycle tire, this group is analyzing how to *ride* a bicycle. After about ten minutes of work, the two groups are told to present their analysis to the rest of the audience. The first group proudly presents their Mager-style tasks and learning objectives and coyly let their professors and colleagues know how much they've learned over the last year or two.

The second group's presentation is a little different. They usually have some rough stick-figure drawings on the board and perhaps a list of pre-requisite skills, but nothing even remotely resembling a clear task analysis or written learning objectives. There is a sense of bewilderment and betrayal on their faces that says, "Nobody ever taught us how to do this," or "I've never seen this in any books I've read." I let them off the hook quickly and introduce the topic at hand. The issue these students run into, as well as many practicing training professionals, is that traditional instructional analysis methods do not account for dynamic, integrated, or concurrent skill sets.

Since the design phase is a natural extension of the analysis, it is very easy to maintain the same static structure of the content. For instance, in the bicycle example above, a traditional task analysis would break down the job into individual tasks such as peddling, steering, keeping balance, and watching road conditions. Learning objectives would then be built to identify the behavior, conditions and standards the learner would be expected to perform (Mager, 1997). Lesson segments would teach the learner the individual skills and associated knowledge with each task (see Figure 1).

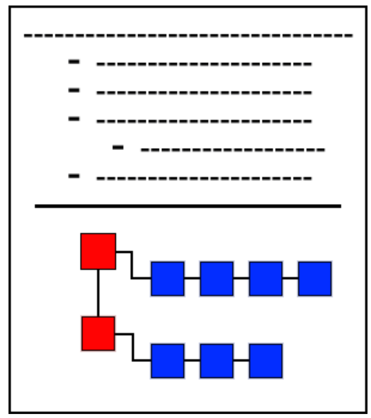

Figure 1. Representation of hierarchal knowledge structures, such as task analysis or chunking.

However, at the end of the day, it doesn't matter how well the student is able to perform the tasks separately, it only matters that they can perform them together.

TRADITIONAL VS MODEL-CENTERED CONTENT STRUCTURES

While not all instructional content necessarily integrates skills, many of today's job requirements involve a high degree of domain knowledge, skills, and problem-solving ability (Berryman, 1993; Jonassen, Hung 2006). The goal of instructional systems design (ISD) is to help learners build usable and appropriate knowledge structures which allow them to perform these job requirements. The ISD process is an approach that guides a designer to systematically create artifacts which will eventually help the learner develop the knowledge structures or mental models (Gibbons, Richards, Nelson, 1999). Therefore, the methods and structures instructional designers use to analyze content have an impact on the design structure and the knowledge structures which the learner builds.

Back to the bicycle example. Obviously, teaching someone to ride a bike using traditional instructional design methods is a little ridiculous. Riding a bicycle is an integrated activity involving several skills (both cognitive and psychomotor) performed simultaneously. The rider must essentially master a model of performance (peddling, steering, watching road conditions, and maintaining balance). Teaching bicycle riding requires strategies such as demonstration, scaffolding and deliberate practice (Ericsson, Krampe, Tesch-Romer, 1993). The strategies support, center on, and enable the performance model: training wheels allow the rider to practice steering and peddling; keeping the rider in a safe area allows gradual learning to deal with different road conditions.

Furthermore, the rider is gaining an understanding of the bicycle itself and controlling speed or direction. The expert performance model is augmented as they begin to experience the relationships between the environment the bicycle is in and how to operate it most effectively. As basic skills are obtained, the rider is introduced to new obstacles, terrain, and road restrictions. Each of these environmental conditions affects how they perform and manipulate the system (bicycle). They are building a cognitive structure which is highly integrated and interrelated (see Figure 2).

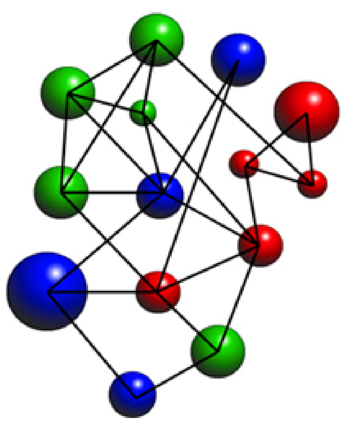

Figure 2. Representation of the interrelationships between expert performance, system, and environments.

Gibbons (2001) identifies this method of training as Model-Centered Instruction (MCI) because it focuses the learner on different models of content (expert performance, cause-effect systems, and environments) and the relationships between them. The use of these models for instruction helps the designer analyze, design and represent knowledge and skills that the learner can hopefully assimilate into their own knowledge structure.

Instructional design theory suggests that the structure and representation of content influences the knowledge structure and representation a learner internalizes and then executes in the real world (Merrill 2002; Dijkstra & van Merrienboer, 1997). The reverse of this applies as well: *if the designer wishes the learner to gain a specific schema or mental model of content, then it is necessary to analyze and represent content into structures similar to that schema or mental model.*

Figure 3 shows a representation of how different structures used to analyze content impact design and presentation of content during training and their probable impact on the mental structures learners develop as a result of the training.

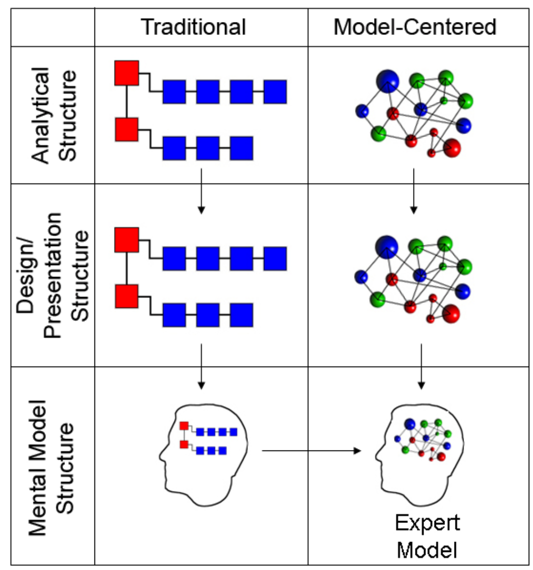

Figure 3. Approaches and structures in the analysis and design phases and their impact on the learner's knowledge structure.

An important element in Figure 3 is in the bottom frames. This representation suggests that if traditional analysis and design methods are used in training, it will likely result in segmented and discrete knowledge. However, if the real-world performance does requires an integrated understanding of the performance, systems and environment, then there is an additional period beyond the allotted learning time required for the learner to translate or restructure their knowledge to match the expert's model.

This restructuring period is commonly observed as students transition from the classroom to the job, especially if practice was not part of the training requirements. Suddenly, knowledge and skills that were learned separately must now be used together. What often results is frustration and cognitive overload until the knowledge patterns are built for more efficient recall.

It stands to reason, then, that if real-world performance requires executing an expert performance model (integrated skills, cause-effect relationships), then a model-centered approach to the training is the most efficient way to help learners develop the knowledge structure they need to perform.

Understanding how to develop model-centered instruction, however, does not come easily. Not all training situations are as clear cut as the bicycle example. Furthermore, instructional designers who have been steeped in traditional ISD methods may find it difficult at first to see the correlations between modeling and traditional approaches. Model-centered instruction has implications at all levels of the design process and designers must develop new ways of looking at content and instruction. This paper, however, will focus only on modeling approaches during content analysis and the impact those models have on design structures.

TASK ANALYSIS

Traditional Task Analysis

The first aspect of model-centered instruction is modeling content at the analysis stage. To understand this process though, it is important to review standard approaches, such as task analysis. MIL-HDBK-29612-2a defines a task as:

> …A single unit of specific work behavior, with clear beginning and ending points, that is directly observable or otherwise measurable. A task is performed for its own sake, that is, it is not dependent upon other tasks, although it may fall in a sequence with other tasks in a mission, duty or job (para. 6.7.1).

From this definition, we see that tasks are intended to be discrete in nature. This is necessary to break down and identify critical elements of the performance without creating "a list of tasks and subtasks far too detailed and lengthy for instructors to provide to students as a basis for preparing them for mission or job performance" (para. 6.7.1.1). The outcome of task analysis, however, produces several artifacts, such as:

> …[a] list of all *tasks*, the *equipment* or materials involved with each task, the *conditions* under which the tasks must be performed, and the standards that must be met (para. 6.7.2, emphasis added).

The artifacts of this analysis identify tasks and standards (expert performance), equipment (cause-effect systems), and conditions (environments). These are important elements because they constitute the actual content of the training, regardless of the way they are phrased or structured. The artifacts of task analysis are then transformed into learning objectives.

Role of Learning Objectives

Learning objectives are a direct result of task and content analysis. They attempt to capture the performance, system and environment elements into a succinct statement of what the learner should be able to do after training. Learning objectives are used to frame assessment activities. In addition to identifying audience and evaluation criteria, learning objectives state a behavior performed under a set of conditions (Mager, 1997). In other words, they describe a specific performance for a system within an environment. For example:

- Operate (performance) the M16A1 machine gun (system) while under simulated-fire conditions (environment).
- Recall (performance) the effects of wet and cold weather (environment) on the human body (system) over time.
- Resolve concerns (performance) with a customer (system) when receiving returned and defective merchandise (environment).

As with traditional task analysis, these learning objective statements are discrete and static in nature. What is difficult to say is whether the structure of the learning objective causes the analysis to be static or whether the analysis forces the learning objective to be static. Either way, the output of traditional ISD analysis resembles anything but dynamic, integrated performance.

Many arguments exist about the use of learning objectives for both designers and students. Some argue that learning objectives are helpful as advanced organizers for preparing learners for what they are about to learn (Ausubel, 1960). Others suggest that stating learning objectives to the student upfront is about as useful as telling a teenager they are about to be lectured for doing something wrong ("Son, we're going to have a little talk about what you did and you need to listen." "Whatever, Dad, just tell me what my punishment is and let's get it over with.").

Despite the exposure learners have with learning objectives, they are of most value to the educator or instructional designer. Learning objectives are closely tied to the content analysis and should reflect what was learned in that phase of the ISD process. Learning objective essentially do four things:

- Identify the actual **content** to be taught (environment, systems, performance)
- Identify the **scope** of the content (environmental conditions, system elements)
- Identify **evaluation** criteria and standards for assessment (observable performance)
- Provide **traceability** to the analysis of the content.

Learning objectives, however, only convey a slice of the content identified in the analysis phase. This mainly has to do with the evaluation requirement for a single, observable performance (see Figure 4). It is not realistic to expect the learner to perform every permutation of the performance, the system, and the environment during the training.

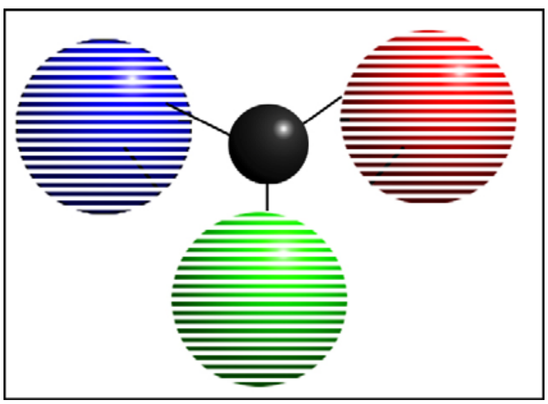

Figure 4. Representation of how learning objective statements only reflect small slices of the performance, system, and environmental content.

Gestalt learning theory suggests that several slices of the content are sufficient to constitute learning (Moore, Fitz, 1993). But in reality, real-world performance requires the person to perform under a wide variety of conditions, not just those assessed in the training. These slices, or discrete learning objective and task analysis structures, only approximate what truly must be learned and applied in the real world. Therefore, it is important that the learner have some type of mental model for how to perform with a system under all the conditions they will likely encounter. This requires a structure which integrates the individual slices into a whole picture.

DYNAMIC CONTENT STRUCTURES

History of Integral Calculus

The discrete nature of learning objectives and traditional task analysis has an interesting parallel in mathematics history. As far back 450 B.C., mathematicians such as Antiphon and Archimedes had basically solved the issue of finding the area of any polygon (O'Connor, Robertson, 1996). The only problem left was how to solve the area of a curved shape. Little did they know it would take almost two millennia to solve.

Their initial approach was to find the area by adding up the values of known polygon shapes that fit into the basic curve shape. They determined that by increasing the number of polygons and using smaller and smaller shapes, the closer the approximation would be to the actual value (see Figure 5).

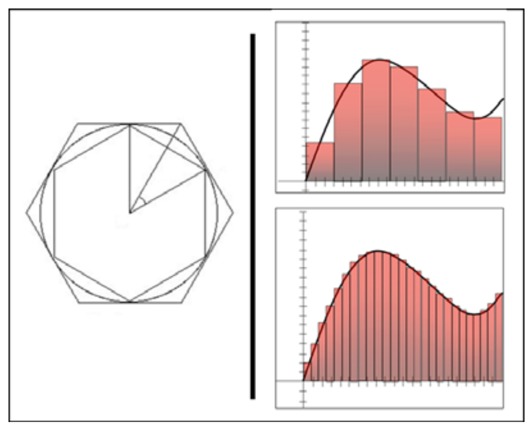

Figure 5. Example of Archimedes method of exhaustion for a circle (left). Approximations of finding the area of a curve (right).

This became known as the Method of Exhaustion because, in simplified terms, the mathematician exhausts all the practical values for the shape…or until they collapsed from exhaustion. Visions of *Last Mathematician Standing* come to mind. In the end, though, their value was still only an approximation of the actual curved area.

The solution revealed itself in the 1700s with the work of Sir Isaac Newton and Leibnitz (O'Connor, et al, 1996). They essentially focused on the characteristics and rate of change of the curve, rather than worrying strictly about the area under it (see Figure 6). By looking at the problem differently, they were able to integrate all of the discrete values into a manageable and more accurate structure. This structure was a formula which accounted of all the possible values within a given range but without the complex addition necessary with the Method of Exhaustion. The range values allowed the mathematician to set boundaries on the scope of what was being calculated.

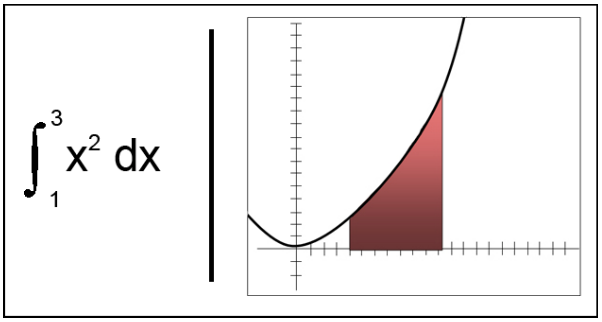

Figure 6. Representation which integrates the complete area under a curve.

Now how does this relate to task analysis and learning objectives? The act of writing endless learning objective and task analysis statements is the equivalent to using the Method of Exhaustion. An analyst can create more and more finite task statements and never actually capture the essence of the whole performance. Task statements may literally identify all the permutations associated with the performance, system, and environment, and result in hundreds of individual "slices" of the content. The details reach the level of being ridiculous and become more than the designer can reasonably manage…until they collapse from exhaustion.

What is required is a different structure for describing the performance, one that looks more carefully at the characteristics and relationships between the performance, system, and environment.

Integrated Performance Structures

Gibbons, Nelson and Richards propose a solution to this problem called the Model-Centered Analysis Process (MCAP) (1999). Originally developed for the Idaho National Laboratory, this process essentially places the content elements (expert performance, cause-effect systems, and environments) and places them into an algorithmic statement or syntactic string (see below).

> In <night> time conditions with <20> knot <headwind> at <35,000> feet, heading <270> degree traveling at <550> knots <indicated>, the aircraft < <radar> shows <two> <unidentified> contacts <50> miles away at bearing <205> degrees on a heading of <035> degrees >. To <intercept> <unidentified> contacts, the <pilot> <increases> <airspeed>, changes <heading>, and <lower> <altitude>. (Hadley, Gibbons, Richards, 2003, p. 5)

This example is an analysis modeling language which captures performance, system, environment and scenario information into a single structure (Gibbons, et al, 1999, p. 22). Its purpose is to allow the designer to create a statement which encompasses all the variables of content. What is important, though, is that the statement is simple in its construction and more content may be added by increasing the range of the variables. This structure is similar to the integral calculus example because it reflects the single formula with defined range values.

Merrill also identifies a method for showing how performance is tied to systems (2002). This method is a step beyond Human Performance/Factors analysis because it takes into account the effect a performance has on a system. Merrill introduced Process Entity Activation Networks (PEANets) as a way of representing, visually, the processes involved within a system and how performance affects those processes (see Figure 7).

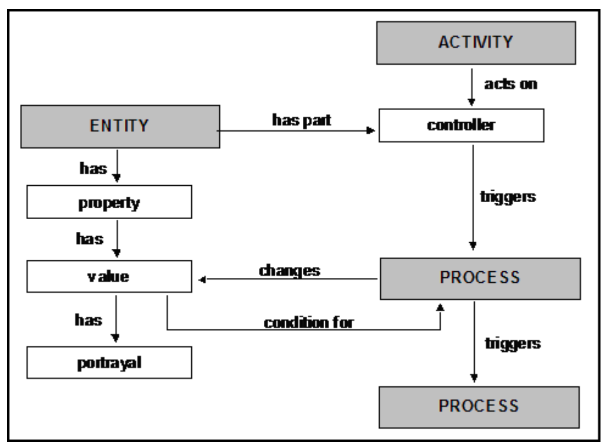

Figure 7. Merrill's Process Entity Activation Network (PEANet) diagram (2001, p. 13).

Merrill's PEANet is an effective way to model performance and cause-effect systems and was intended as a simulation framework in the design and development phase. The benefit of the PEANet is that it exposes specific "if-then" relationships which are typically lost in traditional task analysis. These "if-then" statements tie together the slices of content into a more coherent mental model the learner can use after training. Merrill proposes that "…careful analysis of subject matter content (knowledge) can facilitate both the external representation of knowledge for purposes of instruction (knowledge objects) and the internal representation and use of knowledge by learners (mental-models)" (p. 1).

KNOWLEDGE STRUCTURES AND FIDELITY

Clark and Estes suggest that the differences between knowledge structures or mental models of experts and novices are vastly different.

> Experts are capable of a great variety of advanced skills, including consistent, extremely rapid, accurate and effective diagnosis and solving of complex problems within their domain of expertise. While novices experience a severe limitation on conscious processing… (1996, p. 2)

Obviously, experience and practice play a huge role in explaining this disparity between expert and novice behavior. While the expert and novice may have a similar understanding of facts, concepts and procedures for their domain, the expert seems to have a much stronger understanding of the inter-relationships between them.

Additionally, the expert's knowledge appears to extend beyond a mere "if-then" understanding, such that they can synthesize relationships to troubleshoot or create innovative solutions. This suggests that the fidelity of the expert's mental model of the performance, system and environment more closely represents the real-world model.

Mental Model Fidelity

Zeigler, Kim and Praehofer (2000) provide an interesting framework for modeling system fidelity that is relevant to knowledge structures. In their book *Theory of Modeling and Simulation*, they propose a framework for five levels of modeling fidelity, an extension of George Klir's (1969) work (see Table 1).

Table 1. Levels of modeling fidelity

Level	Attributes	Explanation
0	I/O	Input/Output (If, Then)
1	I/O, *f*	Manipulates input by using a function
2	I/O, *f*, I-state	An initial state affects the function
3	I/O, *f*, I-state, RT-state	Time becomes a dynamic variable to the function (run-time states)
4	I/O, *f*, I-state, RT-state, Coupling	Groups of systems are affected by each others inputs and outputs

The following example shows how the fidelity levels might be applied to modeling the ignition and engine systems of the car. The first level, Level-0, is a simple input-output definition for a component or system (turn ignition key--automobile starts). Next. Level 1, refers to the function or transformation of the input as it becomes the output (ignition key closes electrical circuit and ignites gasoline in motor). Initial states of the system are accounted for and added in Level-2 (amount of battery power and gasoline affect the ignition). A runtime state, in Level-3, looks at variables that affect the input, initial state and function over time (changes to electrical power and gasoline level over time will affect engine performance). Finally, Level-4 (also called coupling) accounts for all the variables as they relate to other systems that the main system interacts with (the alternator provides power to the battery; gasoline pump can fill up automobile's gas tank).

Mental models can take on a similar fidelity. With procedural knowledge, for example, the learner may be able to perform an operation, but their knowledge is restricted to a mere stimulus-response relationship (Level 0). The learner may try to deduce the relationship based on one or two instances, but the mental model may be incorrect or incomplete (Hadley, Gibbons, Richards, 2003).

A Level 1 type of mental model would demonstrate that the learner can predict outcomes because they understand the function behind the relationship. The exercise in Figure 8 is an example of how a learner would build a Level 1 mental model.

Use the pattern to predict the next two numbers in the sequence:

3, 4, 6, 10, 18, 34, 66, 130, ...

Figure 8. Example of learner building a mental model with input/output and functional fidelity.

After careful analysis of the pattern, the student begins to recognize a mathematical function at work. For a given number, subtract one and then multiply that number by two [2(x-1)]. Once the learner has acquired the function, they are capable of predicting any number in the sequence, regardless of where it falls. Similarly, an expert's mental model of a system at Level 1 enables them to predict the output of any number of inputs.

Mental models at Levels 2 and 3 required an in-depth understanding of the system and environment. A Journeyman maintainer, for example, knows that functional parts, circuit cards, or fluids (good initial states) are required to keep a system operational. A Master-level maintenance technician, on the other hand, knows how time affects the quality and performance of a system and can intervene at specific points to keep the system fully functional.

Finally, Level 4 knowledge structures explain how different external systems impact the main system. World-wide economic systems are a prime example of this level. Financial analysts who understand how the trade negotiations or war in one country affect the economics in another country definitely manifest a Level 4 mental model.

When a designer decides what level of knowledge the learner will be required to have, their analysis methods must reflect a structure that yields a similar fidelity. That analysis will then facilitate structures for representing the content in such a way that the learner can develop strong knowledge structures. This is done by focusing on the key relationships between the performance, systems and environments.

Components, Attributes, and Relationships

Merrill's PEANet approach (Figure 7) illustrates an action on an object and the characteristics, representations, and relationships related to that object. His approach helps the designer identify essential elements for the training. First, there is an action on a component. Because of this action, the component is now a relevant piece of information in the instruction. Additional analysis shows that the component has certain attributes. These attributes, likewise, have relevance in the instruction and usually appear as facts or knowledge-level content. What is typically left out, as Merrill explains, is the "what happens" aspect of the action (2002, p. 12). If a button is pressed, then a light comes on. This light is a new component with its own attributes and relationships with other components (see Figure 9).

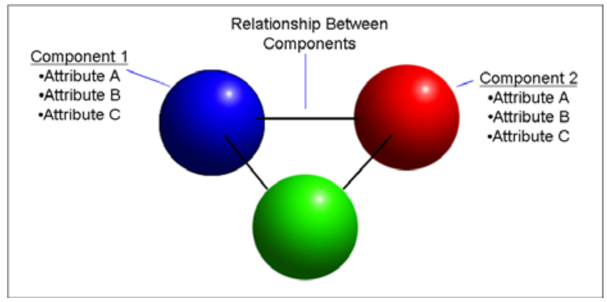

Figure 9. Representation of components, attributes and relationships.

The relationships are a key element to both the analysis and design of the instruction. The modeling fidelity chart (Table 1) is fully concerned with relationships and the degree to which they are represented. As has been shown, the learner's understanding of a relationship is a major factor between novice and expert performance. Unfortunately, relationships are often difficult to articulate, especially in words. In a training environment, special emphasis is made on exposing relationships and allowing learners to gain higher fidelity levels of knowledge.

The following example shows how a traditional examination of content overlooked an important relationship and how reexamining the content from a modeling perspective resulted in a much richer learning experience. This example is taken from a Navy Interior Communications (IC) technician lesson on the cathodic protection systems on ships. One of the cathodic protection methods is called the Impressed Current Cathodic Protection system. This system creates a low-voltage electrical current on the hull of the ship to prevent corrosion. In one section of the lesson, the learner is told that their responsibility is to record the ship's power output and the electrical current level (or potential) on the hull to make sure it is in range. The following text explained what was important for the learner to know:

> The ideal potential for corrosion prevention in salt water is -0.85. Optimum range of polarization on the hull should range between -0.80 and -0.90 volts. When the ship is in fresh or brackish water, more power will be required to maintain this range because the water is less electrolytic than salt water. (Cathodic Protection>Hull Potential>Setting)

Upon reexamination of the content, the instructional designer realized there was a significant relationship between the ICCP system and the water environment. A simple modeling analysis looked at the components of the system and their attributes and identified the relationships between them (see Figure 10).

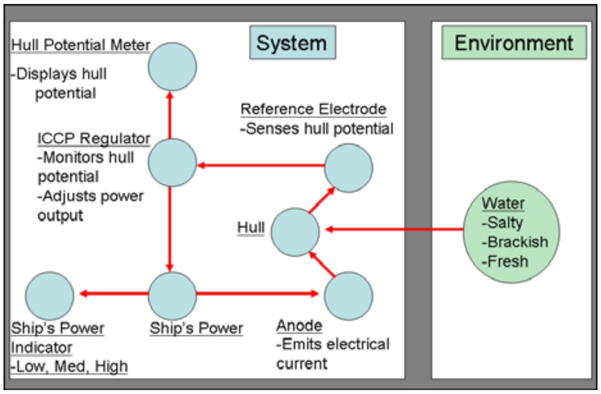

Figure 10. Model of ICCP system to identify relationships.

The dynamic components of the system and environment (controls/indicators) were identified, along with their relevant attributes or variables. The relationships between the component variables were then expressed in a pseudo-code algorithm or syntactic string.

Water values: Salty, brackish, fresh Resistance values: low, normal, high Potential indicator values: -0.90, -0.85, -0.80 Power output values: low, normal, high
When initial state of water is <Salty> then resistance is <normal>, potential indictor equals <-0.85> and power output is <normal>. As water state changes to <fresh>, then resistance is <high> and potential indicator changes <-0.80>. If potential indicator equals <-0.80>, then ICCP regulator changes power output to <high>. If power output equals <high> and water state equals <salty>, then potential indictor changes to <-0.85>.

This relatively simple exercise of modeling the system created the framework for a simulation activity. The activity allowed the student to manipulate the known variables of the environment to see the effect they would have on the ship's power output and the hull potential reading (see Figure 11). The activity reinforced the importance of knowing how water conditions affect the power output and allowed the student to experience the relationship, rather than just read a brief statement about it.

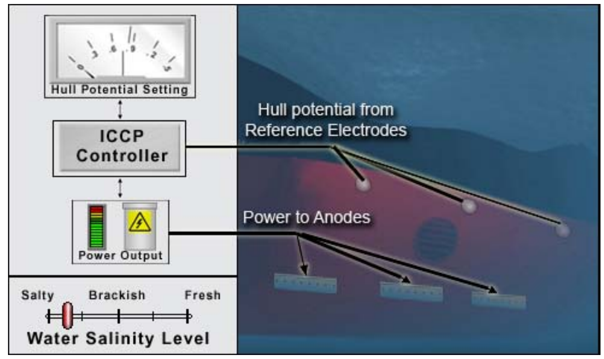

Figure 11. Simulation activity for helping learners recognize the relationship between a system and an environment.

MODELING FOR INSTRUCTION, MODELING FOR SIMULATION

We have addressed the importance of modeling content as a means of identifying critical elements of instruction and creating structures which represent the knowledge structure we are attempting to create in the learner. Instructional designers who are familiar with and capable of using modeling techniques during their analysis discover aspects of the content they would never have identified using static and hierarchical analysis methods. The modeling structures tend to show the pieces and relationships which are missing from their own mental model.

Given the strong emphasis in this paper on modeling techniques for instructional designers, it should by no means imply that they do *all* the modeling for a training project. On the contrary, trainers and instructional designers typically come from educational backgrounds where "math was optional". True simulation modeling requires a strong foundation in mathematics or computer science. It is therefore critical to heavily involve programmers, systems analysts, or computer scientist in modeling and analysis.

As the programmer, instructional designers (ID) and subject matter experts (SME) come together to analyze the performance, systems, and environments, they should be aware of their goals and responsibilities in the modeling process. Figure 12 is a representation of how the three roles interact. In the development of an instructional or training simulation, the three roles must have a shared understanding of the performance, system, and environment. But they use the content differently and structure it in ways that meets their needs.

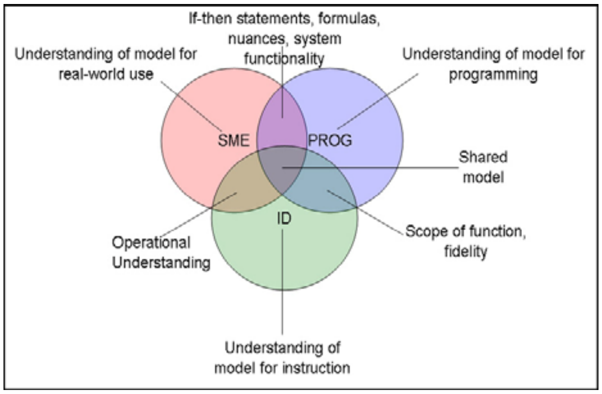

Figure 12. Shared models and understanding of content for different roles.

For example, when analyzing a control panel, the SME may have schematics and electrical diagrams to understand how the panel works. The ID inquires about the procedures the learner must be able to perform. Each task relates to a control or indicator which the ID examines and then identifies the component attributes. Next, the ID articulates the relationship between the component (with input from the SME), its attributes, and other relevant components and attributes in the system ("So what happens when I turn the knob?"). During the process, the ID continues to inquire from the SME which attributes of the components are important to the performance, allowing them to identify the scope and fidelity of the content.

The programmer, who is hopefully listening carefully to the exchange, begins to inquire about additional attributes of the component. They need to know visual and haptic characteristics ("Does it just turn between 1 and 10, or can it move freely 360 degrees? How much resistance does the knob provide when it is turned?"). These issues are beyond the scope of the instructional designer, but they are necessary when creating a realistic learning environment.

As the three roles work together to define performance tasks, system functionality, and environmental impacts, they create a shared model or understanding of the content for training purposes. While they may represent those models in their own way and for different reasons, there are definite overlaps. Often, the designer must consult the programmer for possible methods of modeling the content so is smoother communication when designing training activities. Likewise, the programmer may ask the SME to represent ideas and concepts in formats which are more consistent with the training approach rather than how the SME is used to conveying the ideas. Communication and agreed upon terminology greatly improve the analysis process.

CONCLUSION

As the tasks and equipment individuals use on a daily basis become increasingly complex, the approaches instructional designers use to analyze, design, develop, and evaluate training need to focus on helping the learner develop knowledge structures that match that complexity. While content modeling at the analysis phase is not required or even recommended for every training situation, it is an additional tool instructional designers use to help learners develop a dynamic understanding of performance, systems and environments.

In moving forward, the modeling and simulation field can provide a great deal of guidance for developing modeling methods and approaches for non-technical training professionals. A good amount of work is currently being done in cognitive task analysis, intelligent tutor design, and human-behavioral modeling, but greater attention is required on bringing modeling techniques to instructional systems design.

ACKNOWLEDGEMENTS

Special thanks to the IDs and managers at Sonalysts, Inc. and JHT, Inc. who put up with all my off-the-wall ideas over the years. Thanks for Dr. Gibbons for your genius and friendship. Most of all, thanks to my wife and three kids for reminding me that knowledge and intelligence are best fostered in loving home.

REFERENCES

Ausubel, D. P. (1960). The use of advance organizers in the learning and retention of meaningful verbal material. *Journal of Educational Psychology*, 51, 267-272.

Berryman, S.E. (1993). Learning in the workplace. In L. Darling-Hammond (Ed.), *Review of Research in Education*, vol. 19. Washington, D.C.: American Educational Research Association.

Clark, R. E. and Estes, F. (1996). Cognitive task analysis, *International Journal of Educational Research.* 25(5). 403-417.

Dijkstra, S. & van Merriënboer, J. J. G. (1997). Plans, Procedures,and Theories to Soplve Instructional

Design Problems. In S. Dijkstra, N. M. Seel, F. Schott, & R.D. Tennyson (1997). *Instructional Design: International Perspectives. Volume 2 Solving Instructional Design Problems.* Mahwah, NJ: Lawrence Erlbaum Associates.

Ericsson, K.A., Krampe, R.T., and Tesch-Romer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3), 363-406.

Hadley, J., Gibbons, A., Richards, R., (2003). Plugging instruction into simulation. Paper presented at the 2003 Interservice/Industry Training, Simulation, Education Conference. December, Orlando, FL.

Gibbons, A. S., Richards, R., Hadley, J., Nelson, J. (2003). The design of model-centered instruction: A manual for the design of instructional simulations. Center for Human-Systems Simulation Technical Report, Idaho Falls, ID, Idaho National Engineering and Environmental Laboratory. Summer, 2003.

Gibbons, A.S., Richards, R., Nelson, J. (1999). Model-centered analysis process (MCAP): A pre-design analysis method based on theory and derived criteria. Center for Human-Systems Simulation Technical Report, Idaho Falls, ID, Idaho National Engineering and Environmental Laboratory. Summer, 1999.

Gibbons, A. S. (2001). Model-Centered Instruction. *Journal of Structural Learning and Intelligent Systems*. 14: 511-540.

Jonassen, D.H., & Hung, W. (2006). Learning to troubleshoot: A new theory based design architecture. *Educational Psychology Review*, *18*, 77-114.

Klir, G.J., (1969), *An Approach to General Systems Theory*, Van Nostrand, New York.

Mager, R (1997). Preparing Instructional Objectives. Palo Alto, CA: Fearon Publishers.

Merrill, M. D. (2002). Knowledge objects and mental models. *The Instructional Use of Learning Objects.* Ed. David Wiley. Agency for Instructional Technology.

Method of Exhaustion. Wikipedia. http://en.wikipedia.org/wiki/Method_of_exhaustion. Accessed June 12, 2007.

MIL-HDBK-29612-2a, (1999). Instructional systems development systems approach to training and education, Part 2 of 5.

Moore P. and Fitz C. (1993). Gestalt theory and instructional design," *Journal of Technical Writing and Communication*, Vol. 23, No. 2, pp. 137–157.

O'Connor, J. & Robertson, E. F. (1996). A History of the Calculus. http://www-groups.dcs.st-and.ac.uk/~history/HistTopics/the_rise_of_calculus.html. Accessed June 14, 2007.

Skinner, B.F. (1968). The Technology of Teaching. New York: Appleton-Century-Crofts.

Sweller, J. (1988). "Cognitive load during problem solving: Effects on learning". *Cognitive Science* 12 (1): 257-285.

Zeigler B. P., Kim T.G., Praehofer H. (2000). *Theory of Modeling and Simulation.* Academic Press; 2 edition.