LEARNING OBJECT DESIGN AND SEQUENCING THEORY

by

David A. Wiley, II

A dissertation submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Instructional Psychology and Technology

Brigham Young University

June 2000

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a dissertation submitted by

David A. Wiley, II

This dissertation has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date C. Victor Bunderson, Co-chair

Date J. Olin Campbell

Date M. David Merrill

Date Paul F. Merrill

Date Laurie M. Nelson, Co-chair

BRIGHAM YOUNG UNIVERSITY

As co-chair of the candidate's graduate committee, I have read the dissertation of David A. Wiley, II in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

____________________	____________________________________
Date	Laurie M. Nelson Co-chair, Graduate Committee

Accepted for the Department

____

Paul F. Merrill
Department Chair, Instructional Psychology and Technology

Accepted for the College

____

Robert S. Patterson
Dean, David O. McKay School of Education

ABSTRACT

LEARNING OBJECT DESIGN AND SEQUENCING THEORY

David Wiley

Department of Instructional Psychology and Technology

Doctor of Philosophy

Given the likelihood of the broad deployment of learning objects-based technology, and the dangers of employing it in an instructionally unprincipled manner, the need for an instructional design theory providing explicit support for the instructional design and use of learning objects is clear. “Theory” here follows Reigeluth’s (1999) definition of design theories as “describ[ing] methods of instruction and the situations in which those methods should be used.” This study reviews, synthesizes, and combines four existing instructional design theories, namely Elaboration Theory (Reigeluth, 1999), Work Model Synthesis (Gibbons, et al., 1995), Domain Theory (Bunderson, Newby, & Wiley, 2000), and the Four-Component Instructional Design model (van Merriënboer, 1997) with new work, the result being a new instructional design theory, Learning Object Design and Sequencing Theory (LODAS). LODAS provides guidelines for the analysis and synthesis of an undifferentiated content area (e.g., English), the application of which produces specifications for the scope and sequence of learning objects. The theory also provides a taxonomy of five learning object types and provides design guidance for the different types of learning objects.

Currently, any person or organization that wants to employ learning objects in their instructional design is required to create their own taxonomy of learning objects. The author considers this to be a major cause of the current poverty of practical applications of learning objects. However, taking the taxonomy and learning object design guidelines presented in LODAS, an instructional designer may be able to connect these to the instructional design theory of their choice via the creation of "prescriptive linking material," a considerably simpler exercise than the creation of a new taxonomy. As the theory is tested, this development has the potential to speed the practical adoption of the learning object approach, allow the simplified application of any instructional design theory to the learning object approach, and provide a common ground for future research in the instructional technology called "learning objects."

ACKNOWLEDGMENTS

My thanks go first and foremost to my wife, Elaine, and children, David Enoch and Megumi, who have supported and buoyed me up with love, patience, and understanding throughout my tenure as a graduate student.

I also extend my gratitude to my committee co-chairs, Laurie Nelson, who introduced me to the field of instructional technology, specifically instructional design theory, and taught me to be passionate about it, and Vic Bunderson, who invited me to participate in the development of Domain Theory and first suggested validity argument as a means to test instructional design theories. My gratitude also goes to the entire committee, who read and reread quickly in order to help me finalize the study in a timely manner.

Finally, special thanks go to my external reviewers, Charles Reigeluth and Brandon Muramatsu, for their tireless reading and commenting.

This dissertation was partially funded by the Edumetrics Institute.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Technology is an agent of change, and major technological innovations can result in entire paradigm shifts. The computer network known as the Internet is one such innovation. After affecting sweeping changes in the way people communicate and do business, the Internet is poised to bring about a paradigm shift in the way people learn. Consequently, a major change may also be coming in the way educational materials are designed, developed, and delivered to those who wish to learn. An instructional technology called "learning objects" (LTSC, 2000) currently leads other candidates for the position of technology of choice in the next generation of instructional design, development, and delivery due to its potential generativity, adaptability, and scalability (Hodgins, 2000; Urdan & Weggen, 2000; Gibbons, Nelson, & Richards, in press).

While standards bodies are eagerly developing technological standards for the widespread deployment of learning objects, currently instructional design theories do not explicitly support the design and instructional use of learning objects. As Reigeluth and Frick (1999) have stated, "more [instructional design] theories are sorely needed to provide guidance for … the use of new information technology tools (p. 633)." The purpose of this study is to develop an instructional design theory that provides guidance in the instructional use of learning objects.

This chapter provides a rationale for the current study by presenting an overview of learning objects, the current state of adoption of learning objects, instructional design theory, and the manner in which learning objects can contribute to the success of instructional technology.

What are Learning Objects?

Learning objects are elements of a new type of computer-based instruction grounded in the object-oriented paradigm of computer science. Object-orientation highly values the creation of components (called “objects”) that can be reused (Dahl & Nygaard, 1966). This is the fundamental idea behind learning objects: instructional designers can build small (relative to the size of an entire course) instructional components that can be reused a number of times in different contexts. Additionally, learning objects are generally understood to be digital entities deliverable over the Internet, meaning that any number of people can access and use them simultaneously (as opposed to traditional instructional media, such as an overhead or video tape, which can only exist in one place at a time). Moreover, those who incorporate learning objects can collaborate on and benefit immediately from new versions. This is a significant difference between learning objects and other types of instructional media that have existed previously.

Supporting the notion of small, reusable chunks of instructional media, Reigeluth and Nelson (1997) suggest that when teachers first gain access to instructional materials, they often break the materials down into their constituent parts. They then reassemble these parts in ways that support their individual instructional goals. This suggests one reason why reusable instructional components, or learning objects, may provide instructional benefits. If instructors received instructional resources as individual components, this initial step of decomposition could be bypassed, potentially increasing the speed and efficiency of instructional development.

To facilitate the widespread adoption of the learning objects approach, the Learning Technology Standards Committee (LTSC) of the Institute of Electrical and Electronics Engineers (IEEE) formed in 1996 to develop and promote instructional technology standards (LTSC, 2000). Without such standards, universities, corporations, and other organizations around the world would have no way of assuring the interoperability of their instructional technologies, specifically their learning objects. A similar project called the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) had already started with the financial support of the European Union Commission (ARIADNE, 2000). At the same time, another venture called the Instructional Management Systems (IMS) Project was just beginning in the United States, with funding from Educom (IMS, 2000a). Each of these and other organizations (e.g., ADL, 2000) began developing technical standards to support the broad deployment of learning objects. Many of these local standards efforts have representatives on the LTSC group.

The Learning Technology Standards Committee chose the term “learning objects” to describe these small instructional components, established a working group, and provided a working definition:

> Learning Objects are defined here as any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning. Examples of technology-supported learning include computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, and collaborative learning environments. Examples of Learning Objects include multimedia content, instructional content, learning

> objectives, instructional software and software tools, and persons, organizations, or events referenced during technology supported learning (LOM, 2000).

Adoption of the Learning Objects Idea

There has been a considerable investment in the idea of learning objects. The IMS Project, which develops and promotes compliance with technical standards for online learning, is funded solely by memberships. The highest level of participation is “Contributing Member,” with an annual fee of $50,000, retroactive to the project’s beginning. Over 30 vendors, universities, and other organizations belong to this program (IMS, 2000b) whose membership list reads like a who’s who of software developers and high-powered organizations: Microsoft, Oracle, Sun, Macromedia, Apple, IBM, UNISYS, the U.S. Department of Defense, the U.S. Department of Labor, the California State Universities, International Thompson Publishing, and Educational Testing Service, to name a few. The next level of membership down, the “Developers Network,” has over 200 members, most of which are universities.

Whether or not the learning object paradigm is grounded in the best instructional theory currently available, there can be little doubt that the United States and the world (the ARIADNE coalition has a similar list of European members) are about to be flooded with IMS compliant – in other words, learning object-based – tools. Microsoft has already released a toolset it touts as “the first commercial application of work being delivered by the Instructional Management System (IMS) Project” (Microsoft, 2000). With this broad industrial, governmental, and educational support, it would appear that learning objects may become the technology of choice in the emerging area of online instruction.

A recent report released by investment banking firm W. R. Hambrect contains more than the common predictions for the future of online learning, e.g., that the online learning market will reach $11.5 billon by 2003 (Urdan & Weggen, 2000). As evidenced in the report, even brokers are talking about learning objects and encouraging investors to make sure that the e-learning companies they buy rely on the technology:

> [Online learning content] development cycles are predicted to shorten by 20% every year to two or three weeks by 2004. This imperative will drive more template-based designs and fewer custom graphics. Learning objects will be created in smaller chunks and reusable formats. As a consequence, the industry will become more efficient and competitive…We are convinced that the move to defined, open standards is crucial to the continuing successful adoption of e-learning, especially as it begins to transition beyond early adopters into the rapid growth phase of the market. Authoring tools will need to operate across different platforms and communicate with other tools used to build learning systems. Content and courseware must be reusable, interoperable, and easily manageable at many different levels of complexity throughout the online instructional environment. Enterprise learning systems have to accommodate numerous and varied learner requirements, needs, and objectives. Corporate customers need to be able to easily track content created by multiple content providers through one training management system and search vast local or distributed catalogs of content to identify learning objects or modules on a particular topic. The race for education technology standards is on (Urdan & Weggen, 2000, p.16).

Recognition, adoption, and the potential for future support for the learning objects idea is significant, and includes some of the biggest players in software, higher education, and even investment. Learning objects may become *the* instructional technology of online learning. However, technical standards and venture capital are not enough to promote learning. In order to promote learning, technology use should be guided by instructional principles.

Instructional Design Theory

Instructional design theories have been overviewed frequently in the literature (Dijkstra, Seel, Schott, & Tennyson, 1997; Reigeluth 1983, 1999b; Tennyson, Schott, Seel, & Dijkstra, 1997). Reigeluth (1999) defines instructional design theory as follows:

> [I]nstructional design theories are design oriented, they describe methods of instruction and the situations in which those methods should be used, the methods can be broken into simpler component methods, and the methods are probabilistic. (p. 7)

Reigeluth's current definition of design theory as prescriptive theory follows earlier definitions of design theory (Simon, 1969; Snelbecker, 1974; Reigeluth, 1983). Because the very definition of "theory" in some fields is "descriptive," design theories are commonly confused with other types of theories that they are not, including instructional systems development models, learning theory, and curriculum theory (Reigeluth, 1999). Following Reigeluth and others, "design theory" as used in this study connotes *prescriptive* theory. Instructional design theory, or instructional strategies, guidelines, and criteria for their application, must play a large role in the application of learning objects technology if it is to succeed.

The Role of Instructional Design Theory: Sequence

While groups like the Learning Technology Standards Committee exist to promote international discussion around the technology standards necessary to support learning object-based instruction, and many people are talking about the financial opportunities about to come into existence, there is astonishingly little conversation around the instructional design implications of learning objects.

Indicative of this lack of thought about instructional design is of item 7(d) of the Learning Objects Metadata Working Group’s (a working group of the Learning Technology Standards Committee) Project Authorization Request (PAR) form. The PAR is the mechanism by which IEEE projects are officially requested and approved, and must contain statements of the project’s scope and purpose. Section 7 of the PAR deals with the purpose of the proposed project, and item (d) in the Learning Objects Metadata Working Group’s PAR (LOM, 2000) reads as follows:

> To enable computer agents to automatically and dynamically compose personalized lessons for an individual learner.

As the Learning Object Metadata standard neared finalization in early 2000, some questions were raised regarding the current standard’s ability to achieve this purpose. Apparently no one had considered the role of instructional design in composing and personalizing lessons. At this point a brief discussion of metadata, the focus of the Learning Object Metadata Working Group’s efforts, is necessary.

Metadata, literally “data about data,” is descriptive information about a resource. For example, the card catalog in a public library is a collection of metadata. In the case of the card catalog, the metadata are the information stored on the cards about the Author, Title, and Publication Date of the book or resource (recording, etc.) in question. The labels on cans of soup are another example of metadata: they contain a list of Ingredients, the Name of the soup, the Production Facility where the soup was canned, etc. In both the case of the library book and the can of soup, metadata allow you to locate an item very quickly without investigating all the individual items through which you are searching. Imagine trying to locate *Paradise Lost* by sifting through every book in the library, or looking for chicken soup by opening every can of soup in the store and inspecting their contents! The Learning Objects Metadata Working Group is working to create metadata for learning objects (such as Title, Author, Version, Format, etc.) so that people and computers will be able to find objects by searching, as opposed to browsing the entire digital library one object at a time until they find the right one.

The problem with 7(d) arose when people began to actually consider what it meant for a computer to "automatically and dynamically compose personalized lessons." This meant taking individual learning objects and combining them in a way that made instructional sense, or in instructional design terminology, "sequencing" the learning objects. It seemed clear to a few of us that in order for a computer to make sequencing or any other instructional design decisions, the computer must have access to instructional design information to support the decision-making process. The problem was that no instructional design information was included in the metadata specified by the current version of the Learning Objects Metadata Working Group standard. Once the problem was clearly explained to the other members of the Learning Objects Metadata Working Group, the only viable solution was to edit the PAR and remove 7(d).

The lack of instructional design discussion at this standards-setting level of conversation about learning objects is disturbing, because it might indicate a trend. One can easily imagine technology implementers asking, “if the standards bodies haven’t worried about sequencing, why should we?” Once technology or software that does not support an instructionally-grounded approach to learning object sequencing is completed and shipped to the average teacher, why would he or she respond any differently? This sets the stage for learning objects to be used simply to glorify online instruction, the way clip-art and dingbats are used in a frequently unprincipled manner to decorate elementary school newsletters. Obviously, instructionally grounded sequencing decisions are at the heart of the instructionally successful use of learning objects.

The Role of Instructional Design Theory: Scope

Discussion of the problem of combining learning objects in terms of “sequencing” led to another connection between learning objects and instructional design theory. The most difficult problem facing the designers of learning objects is that of “granularity” (Wiley et al., 1999). How big should a learning object be? As stated above, the Learning Technology Standards Committee’s definition leaves room for an entire curriculum to be viewed as a learning object, but such a large object view diminishes the possibility of learning object reuse. Reuse is the core of the learning object notion, and this is why a more restrictive definition has been proposed for this study. However, because learning objects commonly require the creation of metadata, which means filling out a form of twenty-some odd fields like “Semantic Density” and “Typical Learning Time,” designating every individual graphic and paragraph of text within a curriculum a “learning object” can be prohibitively expensive. From an “efficiency” point of view, the decision regarding learning object granularity can be viewed as a trade-off between the possible benefits of reuse and the expense of cataloging. From an instructional point of view, alternatively, the decision between how much or how little to include in a learning object can be viewed as a problem of "scope." While reality dictates that financial and other factors must be considered, if learning is to have its greatest chance of occurring, decisions regarding the scope of learning objects must also be made in an instructionally grounded, principled manner.

The Role of Taxonomy Development

The discussion of learning object characteristics, such as sequence and scope, leads naturally to the question, "are there different types of learning objects?" In other words, can types of learning objects be meaningfully differentiated? Taxonomy development has historically accompanied instructional design theory development (Bloom, 1956; Gagne, Briggs, and Wager, 1992), and is recommended by Richey (1986) and Nelson (1998) in their instructional design theory development approaches. According to Richey (1986), the development of conceptual models such as taxonomies serves to "identify and organize the relevant variables; defining, explaining, and describing relationships among the variables" (p. 26-27).

While object categorizations exist specific to particular instructional design theories, such as Merrill's (Merrill, Li, and Jones, 1991) sets of process, entity, and activity classifications, a learning object taxonomy compatible with multiple instructional design theories does not exist. The lack of such a broadly applicable taxonomy significantly hinders the application of learning object implementation methodologies to existing instructional design theories, as current practice has been to create theory-specific taxonomies to support each implementation (Merrill, Li, and Jones, 1991; L'Allier, 1998), considerably increasing the effort necessary to employ learning object technologies.

Goals of This Study

Given the likelihood of the broad deployment of learning objects technology, and the dangers of employing the technology in an instructionally unprincipled manner, both of which have been described in this chapter, the need for an instructional design theory that provides explicit support for the instructional design and use of learning objects is clear. Therefore, the purpose of this study is to develop an instructional design theory that (1) provides explicit support for scope and sequence decisions about learning objects, (2) provides a taxonomy of learning object types, and (3) provides design guidance for each type of object. If properly developed, the instructional design theory will not only present a method of instructional design expressible in learning objects, but will present a general taxonomy of learning object types and an example of linking instructional design theory to the taxonomy. The taxonomy and instructional design prescriptions which link the taxonomy to the rest of the instructional design theory should provide others with an example of how to use learning objects with any instructional design theory.

This instructional design theory will be developed through a theory-building process that includes (1) a review of literature focusing on instructional design theories that provide scope and sequence guidance and (2) the development of a taxonomy of learning objects. The literature will be synthesized into a theory of content analysis and synthesis that will end in the content scope specification and sequence specification of learning objects. According to their instructional function as identified in the content analysis and synthesis, these specifications will be linked to a specific type of learning object as identified in the taxonomy of learning object types. Design guidelines for each of the types of learning objects in the taxonomy will be outlined.

Overview of Chapters

This chapter has presented an overview of the background and rationale for this study. Chapter 2 presents the theory-building methodology that will be employed to accomplish the goals identified above. The Methodology precedes the Literature Review in this study because the Literature Review is a key part of the theory-building process. Chapter 3 presents the instructional design literature pertinent to the current study, including a more detailed investigation of instructional design theories that provide scope and sequence guidance and are compatible with learning object-based instruction. Chapter 4 begins with an overview of the manner in which the theory-building methodology has been applied, and ends with the presentation of the instructional design theory – Learning Object Design and Sequencing (LODAS, pronounced “lotus”) Theory. As the purpose of this study is to *build* and not to *test* an instructional design theory, Chapter 5 will present a proposed design experiment framework for testing and improving LODAS.

CHAPTER TWO

THEORY-BUILDING METHODOLOGY

This dissertation builds an instructional design theory that will provide the support necessary for the instructionally sound design and sequencing specifications of learning objects. A theory- or model-building methodology was employed to achieve this end. In this chapter a relatively new instructional design theory-building methodology (Nelson, 1998) will be introduced, and improvements to the Nelson methodology will be suggested. The chapter will close with a brief discussion of the role of theory-testing in theory-building.

Instructional Theory-building Methodology

While instructional technologists and educational psychologists have long engaged in the creation of teaching strategies, few formalized methods exist for the creation of instructional design theory. Nelson (1998) recently reviewed and synthesized the existing methods (Patterson 1977; Reigeluth 1983; Richey, 1986; Snelbecker, 1974; Snow 1971) with experience from case studies in order to create a thirteen-step instructional theory-building process. The Nelson process contains the following steps, each of which is discussed in greater detail below.

1. Define the Purpose of the Theory
2. Select a Paradigm for the Theory
3. Determine the Specific Domain, Situation, or Scope of the Theory
4. Identify an Optimal Process on Which to Model the Theory
5. Develop general criteria for goals, methods, and conditions

6. Develop Goals for the Theory

7. Develop Methods for the Theory

8. Identify Conditions for the Theory

9. Create a Variable Taxonomy for the Theory

10. Finalize the Theory Prototype

11. Formatively Research the Prototype Theory

12. Finalize the Goals, Methods, and Conditions

13. Write Up the Theory

A comparison of the Nelson design methodology with the Waterfall methodology of software design (Royce, 1970; Boehm, 1981), which is structurally similar, showed that Nelson's methodology stands up to all the historical criticisms of the Waterfall method but one. Nelson's methodology sidesteps criticisms regarding the difficulty of making requirements explicit to the software developer, because the "customer" of the instructional theory building process is usually the theory developer. The methodology is also sidesteps criticisms regarding the lateness with which the customer sees the product for the same reason. Finally, the Nelson methodology also sidesteps criticisms of reusability. While individual algorithms and lines of code disappear when a piece of software is compiled, the goals, methods, and conditions of an instructional design theory continue to exist exposed to the world, ripe for reuse.

However, the common weakness of the Waterfall and Nelson methods is their tendency toward linearity. It is nearly impossible for a work as comprehensive and complex as an instructional design theory to be completed successfully in a single attempt. Nelson's method takes this into account by including "Formatively Research the

Prototype Theory" and "Finalize the Goals, Methods, and Conditions" steps, but these imply a single pass through an improvement loop. Accordingly, Nelson's step 12, "Finalize the Goals, Methods, and Conditions," has been changed to "Revisit the Goals, Methods, and Conditions." This change creates an internal improvement loop that feeds back to review initial purposes, goals, conditions, methods, and values as many times as necessary. This improvement loop is driven by primarily conceptual data, such as expert review, as empirical data gathering falls under the purview of theory-testing as opposed to theory-building.

Ongoing improvement is suggested as part of the final step of Nelson's methodology, "Write Up the Theory." A more explicit statement of the need for multiple iterations of the theory-building process to occur, a rigorous theory-testing plan that would provide the data for each design cycle, and an explicit statement like "Implement and Improve the Theory" would strengthen this methodology. Accordingly, the inclusion of an additional step (numbered 13 below) "Plan for Theory-Testing" makes the commitment to ongoing improvement of the theory more explicit, and requires the theory-builder to think forward to the theory-testing process. Another additional step (number 15 below) "Implement and Improve the Theory," articulates the fact that theory building cannot occur in the proverbial ivory tower, but must be grounded in and improved upon by real experience. This approach could be extended to support a full-blown design experiment approach (Brown 1992; Collins, 1990), as recommended by Bunderson, Martinez, and Wiley (2000).

The following description of the theory-building methodology to be used in this study is adapted from Nelson (1998), with a change to step 12 and the addition of steps

13 and 15. (These steps are bolded below to indicate that they have been modified or added to Nelson's methodology.)

1. *Define the Purpose of the Theory* – Delineate the intent of the theory by determining what instructional situation or problem the theory is attempting to address.

2. *Select a Paradigm for the Theory* – The theory developer should determine if the instructional theory will be one intact model, a model with different variations for different conditions, or an independent components model from which an instructor can choose the most salient methods and strategies for his/her purposes.

3. *Determine the Specific Domain, Situation, or Scope of the Theory* – Determine more specific boundaries for the theory. One goal of this part of the theory-building process is to generally define what the scope of the theory is and is not. This provides the foundation for identifying and articulating the specific conditions under which the theory should be used.

4. *Identify an Optimal Process on Which to Model the Theory* – Find an existing case or create a mental model from which to work (as in grounded theory development). This initial model serves as the framework around which the synthesis of the theoretical and research literature will be organized.

5. *Develop general criteria for goals, methods, and conditions* – Develop criteria to assess the appropriateness of goals, methods, and conditions either identified in existing theories and research or newly formulated for this specific theory.

6. *Develop Goals for the Theory* – Determine the specific learning goals the theory is being designed to fulfill.

7. *Develop Methods for the Theory* – Formulate appropriate instructional methods and strategies necessary to realize the learning goals of the theory.

8. *Identify Conditions for the Theory* – Conditions need to provide guidelines by which users of the instructional theory can judge the appropriateness of this approach for their instructional purposes.

9. *Create a Variable Taxonomy for the Theory* – Group together goals, methods, and conditions into a representation that highlights commonalities and relationships (this is useful in conceptualizing the theory and communicating it to others).

10. *Finalize the Theory Prototype* – A prototype is created that outlines the organization and implementation of the methods and strategies to be used during instruction.

11. *Formatively Research the Prototype Theory* – Refine the theory by doing a type of formative research using existing instructional theories and research findings. Examine the comprehensiveness, cohesion, and congruency of the new theory. Expert review is suitable for this stage of research. Appropriate modifications can be made based on these findings.

12. ***Revisit the Goals, Methods, and Conditions*** – Once the usability of the theory has been tested conceptually, the developer then takes the insights gained and revisits the goals, methods, conditions, and values for the instructional theory. This step takes a rapid-prototyping approach in which the results of the

formative research are fed back into the theory-building system both early and frequently in the theory-building process, continuing until a satisfactory set of the goals, methods, conditions, and values exists.

13. ***Plan for Theory-Testing* –** Develop a comprehensive, rigorous plan for testing the instructional design theory. This plan can take the form of a validity argument, relying on Messick's (1995) unified validity framework, and address aspects of relevant descriptive theories (domain and individual difference theories). The test plan should also include both broad and specific measures of the design (prescriptive) theory itself, such as measuring instructional quality through student learning (broad) and measuring the instructional function of specific prescriptions (specific).

14. *Write Up the Theory* – Formalize the current version of the theory by writing it up and disseminating the theory so that it can be examined and critiqued by other educators and theorists. This allows for further refinement of the theory by the developer and others.

15. ***Implement and Improve the Theory* –** Once the first cycle of theory-building has been completed, which includes conceptual testing, the first cycle of theory-testing can begin, which includes the collection and analysis of empirical data. Carry out the first phase of the theory-testing plan outlined in 13 above. Implementation provides valuable formative evaluation of the theory. Data gathered during implementation in accordance with the testing plan provides more data to feed back into the theory-building process, allowing further refinement of the theory.

This methodology will be employed in the development of the proposed theory, which will be presented in Chapter 4. However, the development of an instructional design theory does not end after the first conceptual testing – the theory can be improved through empirical testing.

Role of Theory-Testing Plans

As stated above, a theory-testing plan provides the theory-builder with concrete strategies for verifying the efficacy of the instructional design theory. Such a plan adds significant value to an instructional design theory, both for the developer and implementers of the theory. As the developer thinks through and develops the theory-testing plan, the instructional design theory-building process becomes grounded in the realities of the testing plan. Likewise, when the theory is written up and published, as per step 14 above, theory implementers have specific metrics they can plan to use in evaluating the performance of the theory. Knowing the needed data types ahead of time allows implementers to quickly put the theory and necessary data collection frameworks into place, and get to the business of helping people learn. Two clarifications should be made before proceeding.

First, it should be noted that it is impossible to perform direct empirical tests of the theory *itself*. That is, for the purposes of empirical testing, it is not possible to isolate the prescriptive theory from the implementation of the theory's prescriptions – the two are totally confounded. Empirical tests will always involve an implementation, and to a large extent it is this implementation that is tested in the theory-testing process. Because this is true, a significant number of implementers and implementation cycles are necessary to adequately test, or begin to validate, prescriptive theories. The stable form, factor, or principle component that will begin to show itself over multiple implementation cycles is the theory itself. This is what makes the design experiment framework laid out by Bunderson (2000), in which a method is explained for meaningfully comparing across cycles, so critical. To perform the necessary number of implementation cycles *without* introducing design improvements based on data collected in previous cycles is to miss the only real opportunity to improve the design theory. Consequently, the theory-testing plan must include implementation-testing procedures.

Second, it should also be noted that the theory-testing plan cannot simply be developed, set aside, and brought out again when the theory is ready for implementation. Like the theory-building process itself, the plan must be developed iteratively, over cycles. As the theory-building process progresses, the theory-testing plan may progress as well. The theory-testing plan was therefore developed simultaneously with the new instructional design theory. Because performing actual theory testing is outside the scope of the current study, the theory-testing plan is presented in Chapter 5, Accomplishments and Future Directions.

Summary

This chapter has outlined the manner in which the proposed instructional design theory will be created and conceptually tested. It has also outlined a rationale for theory-testing, within which the new instructional design theory may be improved and validated over time. Chapter 4 will describe the development of the theory and then present the new instructional design theory and Chapter 5 will present the theory-testing plan.

CHAPTER THREE

LITERATURE REVIEW

The purpose of this chapter is to provide an overview of instructional design theories that provide guidance for making scope and sequence decisions. This review is comprised of three main sections. First, the relatively new field of learning objects will be reviewed in order to provide context for the overview of scope and sequence guidance provided by current instructional design theories. Second, a discussion will be presented describing why the instructional design theories chosen for review were selected. Finally, the instructional design theories to be synthesized, adapted, and extended to support learning object-based instruction will be reviewed.

Learning Objects

The IEEE's Learning Technology Standards Committee has presented a definition of "learning object" that will soon make its way into an internationally recognized standard (LTSC, 2000). As noted above, the definition "any entity, digital or non-digital, that can be used, re-used, or referenced during technology supported learning," is extraordinarily broad. Accordingly, different groups outside the Learning Technology Standards Committee have created different terms that generally narrow the scope of the canonical definition down to something more specific. Other groups have refined the definition but continue to use the term "learning object." Confusingly, these additional terms and differently defined "learning objects" are all Learning Technology Standards Committee "learning objects" in the strictest sense. This section of the literature review will overview several definitions of "learning object" and similar terms and provide the working definition of "learning object" to be used throughout the rest of the dissertation.

The proliferation of definitions for the term “learning object” makes communication confusing and difficult. For example, computer-based training (CBT) vendor NETg, Inc., uses the term “learning object” but applies a three-part definition: a learning objective, a unit of instruction that teaches the objective, and a unit of assessment that measures the objective (L’Allier, 1998). Another CBT vendor, Asymetrix, defines learning objects in terms of programming characteristics: “ToolBook II learning objects - pre-scripted elements that simplify programming … provide instantaneous programming power” (Asymetrix, 2000). The NSF-funded Educational Objects Economy takes a technical approach, only accepting Java Applets as learning objects (EOE, 2000). It would seem that there are almost as many definitions of the term as there are people employing it.

In addition to the various definitions of the term “learning object,” other terms that imply the general intention to take an object-oriented approach to computer-assisted instruction confuse the issue further. (An in depth discussion of the precise meanings of each of these terms would not add to the main point of this discussion: the field is still struggling to come to grips with the question “What is a learning object?”.) David Merrill uses the term “knowledge objects” (Merrill, Li, and Jones, 1991). Merrill is also writing a book on the topic of object-oriented approaches to instruction to be called “Components of Instruction” (personal communication, March 21, 2000), which is sure to introduce yet another term, “instructional component,” into the instructional design vernacular. The previously mentioned ARIADNE project uses the term “pedagogical documents” (ARIADNE, 2000). The NSF-funded Educational Software Components of Tomorrow (ESCOT) project uses the term “educational software components” (ESCOT, 2000), while the Multimedia Educational Resource for Learning and On-Line Teaching (MERLOT) project refers to them as "online learning materials" (MERLOT, 2000). Finally, the Apple Learning Interchange simply refers to them as "resources" (ALI, 2000). While each of these is something slightly different, they all fit the Learning Technology Standards Committee's "learning object" definition.

This terminological confusion forces a study that intends to provide instructional design support for learning objects to answer the question, "For the purposes of this study, what is a 'learning object'?" The Learning Technology Standards Committee definition is too broad to be useful for this study, but the creation of yet another term only seems to add to the confusion. While the creation of a satisfactory definition of the term learning object will probably consume the better part of the author's career, a definition must be decided upon for the purposes of this study. Therefore, this study will define a "learning object" as "any digital resource that can be reused to support learning." This definition includes anything that can be delivered across the network on demand, be it large or small. Examples of smaller reusable digital resources include images or photos, live data feeds (like stock tickers), live or prerecorded video or audio snippets, small bits of text, animations, and smaller web-delivered applications, like a Java calculator. Examples of larger reusable digital resources include entire web pages that combine text, images and other media or applications to deliver complete experiences, such as a complete instructional event. This definition of learning object, "any digital resource that can be reused to support learning," has been adopted for two reasons.

First, the definition is sufficiently narrow to define a reasonably homogeneous set of things: reusable digital resources. At the same time, the definition is broad enough to include the estimated 15 terabytes of information available on the publicly accessible Internet (Internet Newsroom, 1999). While the definition must exclude some things, the primary purpose of the learning object paradigm, as described earlier, is to facilitate reuse. To ignore 15 terabytes of existing data would fly in the face of this central purpose.

Second, the proposed definition is based on the LTSC definition (and defines a proper subset of learning objects as defined by the LTSC), making issues of compatibility of learning object as defined within this study and learning object as defined by the LTSC explicit. The proposed definition captures the critical attributes of a learning object, “reusable,” “digital,” “resource,” and “learning,” as does the LTSC definition. However, the proposed definition differs from the LTSC definition in two important ways.

First, the definition explicitly rejects non-digital (by dropping the word and dropping the idea of a learning object being simply "reference"-able) and non-reusable (by dropping the phrase "used or" which seems to imply the acceptance of single use) resources. The definition of learning object adopted for this study does not include actual people, historical events, books (in the traditional sense of the term), or other discrete, physical objects. The definition also drops the phrase "technology supported" which is now implied because all learning objects are digital.

Second, the phrase "to support" has been substituted in place of "during" in the LTSC definition. Use of an object "during" learning doesn't connect its use to learning. The LTSC definition implies that nothing more than contiguity of and object’s use and the occurrence of learning is sufficient. The definition adopted for this study emphasizes the purposeful use (by either an instructional designer, an instructor, or a student) of these objects to support learning.

With the term learning object defined for this study, the discussion of the instructional use of learning objects can proceed.

Selection of Instruction Design Theories for Inclusion in this Review

An instructional designer charged with creating learning objects and instruction based on them would probably operationalize the question "What is a learning object?" by asking questions like "What are the components of a learning object? What is a learning object made of? What goes into a learning object? How much of it goes in?" These are questions of *scope*. Once these questions had been answered satisfactorily, the instructional designer would probably ask questions like "So what do I do with these learning objects? What can they be used for? How do I arrange them? In what order should the learner encounter them?" These are questions of *sequence*. In other words, the first two questions that must be answered regarding the design and combination of learning objects are questions of scope and sequence. Therefore, the primary goal of the proposed instructional design theory is to provide scope and sequence guidance for the design of learning objects. In Reigeluth's (1999a) terminology, the primary goal of the theory is to significantly increase (1) the efficiency with which instruction can be created and (2) the effectiveness of learning objects-based instruction by providing instructional design support for the creation of learning object-based instruction. Reigeluth's third metric of appeal is not addressed directly in this first version of the theory, as significant increases in the appeal of the instruction may correlate highly with adaptive features of the learning object-based instruction (Reigeluth, personal communication, June 14, 2000). Adaptivity to individual differences (such as learning style) has consciously been scoped out of this first version of the theory; however, once considerations of adaptivity are combined with the learning object approach, it should be considerably more appealing to students who are consumers of the instruction.

Accordingly, instructional design theories that provide overt guidance for making scope and sequence decisions were selected for review and used in creating the proposed instructional design theory for learning objects.

Review of Selected Instructional Design Theories

Four compatible instructional design theories were identified as containing overt scope and sequence guidance that could be applicable to learning objects: Elaboration Theory (Reigeluth, 1999a), Work Model Synthesis (Gibbons, et al., 1995), Domain Theory (Bunderson, Newby, & Wiley, 2000), and the Four-Component Instructional Design model (van Merriënboer, 1997). These theories are reviewed below. Each review contains a description and discussion of the theory, ending with a summary of the scope and sequence guidance provided by the theory. These guidelines for scope and sequence will be synthesized into a single set of compatible guidelines in Chapter 4 as part of the proposed theory-building activity.

Elaboration Theory. Reigeluth's (1999a) Elaboration Theory helps users "select and sequence content in a way that will optimize the attainment of learning goals" (p. 426). It has been described numerous times in the literature (Reigeluth, et al. 1980; Reigeluth & Darwazeh, 1982; Reigeluth, 1992, 1979) and has been critically reviewed by Wilson and Cole (Wilson & Cole, 1992).

The theory's Simplifying Conditions Method (SCM) provides relevant information regarding scope and sequence of instructional content. SCM is composed primarily of two parts, epitomizing and elaborating. *Epitomizing* means finding the simplest version of the task that is to be taught that is still representative of the entire task. *Elaborating* means teaching students increasingly complex versions of the task. The idea of incrementing the difficulty of instruction has been discussed in other ways, such as Bruner's (1960) "spiral curriculum," Ausubel's (1968) "progressive differentiation," and Burton, et al.'s (1984) "increasingly complex microworlds." Reigeluth (1999a) described epitomizing and elaborating as follows.

> Epitomizing utilizes
>
> 1. a whole version of the task rather than a simpler component skill;
> 2. a simple version of the task;
> 3. a real-world version of the task (usually); and
> 4. a fairly representative (typical or common) version of the task (p. 444).
>
> Elaborations should be
>
> 1. another whole version of the task;
> 2. a slightly more complex version of the task;
> 3. equally authentic (or more so); and
> 4. equally or slightly less representative(typical or common) of the whole task (p. 444).

The SCM can be summarized in the following nine steps. (Detailed guidelines for the design of an SCM sequence are available in Reigeluth, 1999a.)

1. Prepare for the content analysis and instructional design.
2. Identify the simplest version of the task to be taught, paying careful attention to the simplifying conditions (i.e., the conditions which make this version of the task simpler than others).

3. Analyze the organizing content for this task. (This is called "organizing content" because different organizational strategies are presented for procedural, heuristic, and tasks containing a combination of the two).
4. Analyze the supporting or prerequisite content.
5. Decide the size of the individual instructional episodes. "Too big is bad…Too small is bad" (p. 447). Appropriate size is situational, and varies depending on delivery constraints (such as time, learner ability, content difficulty, etc.) Episodes need not be of equal size.
6. Determine within-episode sequencing of the content.
7. Identify the next version (first elaboration) of the task.
8. Analyze organizing content, supporting content, and determine size and within-episode sequencing of content (steps three - five) for the next version of the task.
9. Cycle back to step seven to identify the remaining versions of the task and design the instruction for each.

While Elaboration Theory and its Simplifying Conditions Model were not created with learning objects in mind, they do provide explicit guidance for both scope and sequence for developing instruction. This guidance can be adapted to learning objects as follows.

Scope guidance summary: Learning objects need not all be the same size; while some will be small, they should be combined into learning objects large enough to teach either the epitome or current elaboration.

Sequence guidance summary: Learning objects should be presented in order of increasing complexity, beginning with the epitome or simplest case.

Work Model Synthesis. In many ways, Work Model Synthesis (Bunderson, et al., 1981, Gibbons, et al., 1995; Gibbons and Fairweather, 1998) represents a reaction against the fragmentary nature of traditional “analyze down to objectives” approaches to instructional design, such as task analysis. Gibbons, et al. (1995) described Work Model Synthesis as “systematically combin[ing] and recombin[ing] tasks and objectives that through task analysis procedures have been fragmented at a low level.” Gibbons, et al. (1995) catalog problems (from the designer’s point of view) with the traditional, strict one-to-one mapping of objectives to instructional events. These include the need for instruction and practice to occur in different media, the need for practice in increasingly realistic performance environments, the need for more integrative performances, and the need to group related objectives. Work Model Synthesis also builds on Burton, et al.’s (1984) notion of “increasingly complex microworlds”.

Work Model Synthesis is described as a mapping construct (Gibbons, Bunderson, Olsen, & Rogers, 1995) or mapping entity (Gibbons & Fairweather, 1998). In other words, Work Model Synthesis provides a framework in which individual objectives can be combined into meaningful, real world performances, or work models. When constructed properly, the results of work models are valuable performances that people can imagine themselves doing in the real world, as opposed to narrow, individual objectives that frequently hold little or no apparent relevance for the learner.

A work model is *not* an instructional event. Rather, it is a specification from which several individual instructional events can be created. Several types of mappings from objectives to work models are possible. These include:

- Many-to-one mapping, in which many instructional objectives are integrated into a single work model. For example, the objectives “Use capitalization properly,” “Use punctuation properly,” and “Spell words correctly” may be combined into a “Write an admissions essay” work model. As opposed to the individual objectives, performance of the work model has value implications and many students will be able to imagine themselves engaging in the activity, adding relevance to the task.
- One-to-many, in which a single instructional objective maps into multiple work models. For example, the objectives above may additionally map into the work models “Create a resume,” and “Submit an article for publication.”
- One-to-one mapping, in which an instructional objective maps directly into a single work model. (This type of mapping occurs when a prerequisite skill must be mastered in isolation before it can be combined into a more meaningful work model.) For example, the objective “Use word processing software effectively” could be practiced in isolation from the other work models listed above that rely on this mastery.

Because work models are specifications for instructional events, and not actual instances of them, they provide the blueprint for several distinct but similar instances. It is therefore possible that work models could be used to create a number of *instructionally equivalent* events that could be traversed (or by-passed) in different orders by different learners. Gibbons and Fairweather (1998) refer to this possible set of events as "the complete field of problems" that learners could traverse on their way toward expertise, and dub this type of approach the "field theory of curriculum design." Interestingly, this ability to present different learners *equivalent but different* instructional events points toward the possibility of adaptive instruction based on a computer adaptive testing model, in which instructionally equivalent but presentation-style diverse learning objects (visually-oriented, simulation-oriented, etc.) could be delivered based on learner profiles.

The Work Model Synthesis approach provides explicit guidance for both scope and sequence that can be adapted to learning objects as follows.

Scope guidance summary: Learning objects are the instructional events resulting from the instantiation of a work model, and must be large enough to teach meaningful, real-world performances. One or more traditional instructional objectives may map into a learning object.

Sequence guidance summary: Learning objects should be sequenced in an order that simulates the real-world performance with increasing fidelity. Because more than one learning object can be created from a single work model, and because the learning objects thus produced will function in instructionally equivalent ways, instructionally equivalent learning objects can be substituted for each other in the sequence.

Domain Theory. Domain Theory (Bunderson, Newby, & Wiley, 2000) can be viewed as an extension of the Work Model Synthesis approach applied to assessment. Domain Theory provides scope and sequence guidance grounded in fundamental measurement theory. Domain Theory provides a rigorous method for exploring and mapping a domain of expertise, relying on four mathematically sophisticated types of invariance to provide a stable view of the domain over time: sample invariance, task invariance, unit of measurement invariance, and interpretive invariance. This primary focus of Domain Theory, construct valid measurement, is outside the scope of this theory-building exercise and is therefore not discussed in detail here. However, one of the analysis techniques employed in the domain mapping method provides scope and sequence guidance, and this technique will be reviewed here.

While Work Model Synthesis directs users to group related individual objectives into work models, little formal support is given for the manner in which this is to be done. Building on the performance-based practice of Work Model Synthesis, Domain Theory moves computer-assisted instruction toward *expertise representation*, that is, Domain Theory works to represent what people can *do* with their knowledge rather than to abstractly represent what they *know*. Domain Theory takes the work model specification produced by Work Model Synthesis and examines the underlying constructs of expertise within the given domain. For example, in the domain of language acquisition, the underlying constructs may be categorized into the groups listening, speaking, writing, reading, and word knowledge (Strong-Krause, 2000). These groupings may be considered dimensions of expertise within the domain. Domains may have different numbers of dimensions of expertise, and the names of these dimensions are likely to change from domain to domain. These dimensions are analogous to factors in the factor analysis procedure of statistics.

Once these constructs of expertise are identified, they can be considered dimensions that define the expertise space being modeled, just as the traditional four dimensions (height, length, width, and time) define “real” space. In this metaphor, the point of origin represents no expertise in the domain, and further distances from the origin represent greater degrees of expertise. For example, in the language acquisition domain, the task “introduce yourself to a stranger” is closer to the origin of the *speaking* dimension than “make a presentation without notes.” The individual tasks or performances that make up the domain can be located along each of these unidimensions. This location exercise provides a rough ordering of task difficulty in the domain, similar to the idea of increasingly complex microworlds.

Because of its roots in measurement explained above, Domain Theory also elaborates the role of assessment missing from Work Model Synthesis, by making explicit the fact that the same work models that provide specifications for instructional events can also be used to create assessments. This provides an integrated view of instructional design and development in which a single specification entity is created from which both instruction and assessment can be instantiated.

As with the other theories, Domain Theory was not created to provide explicit support for making scope and sequence decisions about learning objects. However, its scope and sequence recommendations may also be extrapolated to support the design of learning object-based instruction.

Scope guidance summary: Learning objects are the instructional events *and assessments* resulting from the instantiation of a work model, and must be large enough to teach meaningful, real-world performances. The scope of individual objects increases as the object’s distance from the point of origin on a unidimensional expertise scale increases.

Sequence guidance summary: Learning objects should be sequenced according to their difficulty order on the unidimensional expertise scales (multiple, non-identical unidimensions (scales) identify a multidimensional space). Because the difficulty of some learning objects may be indistinguishable from others (in terms of placement on the scale), difficulty-equivalent objects can be substituted for each other in the sequence in a manner similar to computer adaptive testing.

Four-Component Instructional Design (4C/ID) model. Van Merriënboer's 4C/ID model of instructional design (van Merriënboer, 1997; Paas & van Merriënboer, 1992; van Merriënboer & Dijkstra, 1996; van Merriënboer, Jelsma, & Paas, 1992) outlines a sophisticated manner of designing training that supports the learning of complex cognitive skills.

As the name suggests, the 4C/ID model has four major steps. These are:

1. "Principled skill decomposition," or breaking the complex cognitive skill to be trained into a set of recurrent (algorithmic) constituent skills and a set of nonrecurrent (heuristic) constituent skills;
2. Further analysis of these two sets of constituent skills revealing the knowledge that supports these skills,
3. Selecting instructional methods for practicing constituent skills and presenting supporting information (notice that the emphasis is more on active practice than information presentation); and
4. Composing a training strategy (van Merriënboer, 1997).

Van Merriënboer's theory is grounded in a significant amount of research (for examples, see van Merriënboer, J. J. G., 1990a, 1990b; van Merriënboer & De Croock, 1992; van Merriënboer , Jelsma, & Paas 1992; van Merriënboer & Dijkstra 1996; Paas & van Merriënboer, 1992, 1994a, 1994b) and provides very specific support in terms of instructional strategies for teaching the two types of constituent skills. Recurrent constituent skills, or skills that are performed in the same manner each time, are taught using part-task practice and prerequisite knowledge is provided just-in-time (van Merriënboer, 1997; van Merriënboer & Dijkstra, 1996). For example, "adding three digit numbers" is performed the same way every time, making it a recurrent skill. This skill may actually be comprised of two sub-skills, "adding one digit numbers" and "carrying." Part-task practice would recommend that each of these component skills be taught and practiced in isolation until automaticity is achieved. Any information the learner needed to perform these tasks would be readily available during practice, when needed.

Nonrecurrent skills, or skills which are performed differently in different situations, are taught using whole-task practice and their supporting knowledge is presented in ways that promote elaboration and understanding (van Merriënboer, 1997; van Merriënboer & Dijkstra, 1996). For example, computer programming would generally be classified as a nonrecurrent skill. The supporting knowledge for this skill, such as the different loop structures, variable types, etc., is taught ahead of time in an elaborative manner so that the learner will have it directly available (from their own memory) during practice. Whole-task practice means that this skill would be taught with a holistic approach such as the Simplifying Conditions Model.

Van Merriënboer identifies three levels of scope, or granularity: skill clusters, case types, and specific problems. Once the constituent skills have been identified, behavioral objectives indicating exit behaviors, etc., can be created for each constituent skill. Van Merriënboer would prefer a method like Elaboration Theory (Reigeluth, 1999a) for re-integrating the skills before teaching them; however, he notes that some tasks are so complex as to escape epitomization at a sufficiently simple level. Instead he calls for building two to five clusters of related skills (*skill clusters*). These clusters represent whole tasks, and comprise the highest level of granularity in the 4C/ID model. Nested within each of these skill clusters is a number of *case types*. Case types are the categories of problems and examples the learner will experience during the instruction, and represent the middle level of granularity within 4C/ID. Finally, within each case type there are several *specific problems*. These are the actual examples and problems with which the learner interacts during instruction, and represent the lowest level of granularity within the 4C/ID model.

Van Merriënboer also identifies three levels of sequencing: macro-level, meso-level, and micro-level. The 4C/ID model provides detailed directions on sequencing at each of these levels. At the *macro-level,* skill clusters are ordered according to a part-task sequence. These clusters should be ordered such that the skills in the first cluster are “prerequisite” for success in the second cluster, and so on. At the *meso-level,* case types are ordered according to a whole-task sequence. Finally, at the *micro-level,* the 4C/ID model presents sequencing options for specific problems based on Sweller’s (1988) interpretation of Cognitive Load Theory, in which problem formats (or types) and sequence interact. For complicated tasks where learner interaction with examples and problems is expected to produce high cognitive load (e.g., conventional problems with no performance constraints), case types are ordered in the standard simple to complex manner to avoid cognitive overload. However, in order to promote transfer of skills to novel problem-solving situations, the 4C/ID model recommends finding low load problem formats (e.g., worked-out examples, completion assignments, and conventional problems with no performance constraints) and varying the problem sequence randomly (van Merriënboer, 1997).

As with the first three theories, the 4C/ID model was not created to provide explicit support for making scope and sequence decisions about learning objects. However, its scope and sequence recommendations may be extrapolated to support the design of learning object-based instruction.

Scope guidance summary: Learning objects can be of two sizes: skill clusters (macro-level) and specific problems (micro-level). Skill clusters should be scoped so that a single cluster requires no longer than 200 hours to learn (van Merriënboer, 1997). The first cluster should be small enough for learners to begin practicing a simplified, but authentic, version of the whole task within the first few days. The final cluster must be large enough to rely on all of the constituent skills identified in the preliminary analysis. Specific problems should only be large enough to provide examples or practice of a specific skill.

Sequence guidance summary: Learning objects should be sequenced according to their level and type, and in order to promote transfer when feasible. Macro-level skill clusters should be sequenced in a part-task manner, meaning that skills are taught one at a time and gradually combined. Meso-level case types should be sequenced according to a whole-task order, in which all skills are taught simultaneously. Micro-level specific problems can be sequenced in common simple-to-complex order or, when feasible, in a random sequence in order to promote transfer.

Summary

Four instructional design theories have been reviewed: Elaboration Theory, Work Model Synthesis, Domain Theory, and the Four-Component Instructional Design model. Each is compatible with the use of learning objects in computer-assisted instruction, although the authors did not make this compatibility explicit, since learning objects did not exist as such when the theories were created. Each provides guidance for scope and sequence decisions from different points of view, such as measurement theory and complex cognitive skills training, and with varying degrees of specificity. The principles contained in these theories and identified in this review will provide the basis for a more focused instructional design theory tailored specifically to supporting scope and sequence decisions in the design of learning objects. These principles may be summarized as follows:

<u>Scope</u>

- Learning objects need not all be the same size; while some will be small, they should be combined into learning objects large enough to teach either the epitome or current elaboration (Elaboration Theory).
- Learning objects are the instructional events resulting from the instantiation of a work model, and should be large enough to teach meaningful, real-world performances. One or more traditional instructional objectives may map into a learning object (Work Model Synthesis).
- Learning objects are the instructional events *and assessments* resulting from the instantiation of a work model, and should be large enough to teach and assess meaningful, real-world performances. The scope of individual objects

increases as the object's distance from the point of origin on a unidimensional expertise scale increases (Domain Theory).

- Learning objects can be of two sizes: skill clusters (macro-level) and specific problems (micro-level). Skill clusters should be scoped so that a single cluster requires no longer than 200 hours to learn. The first cluster should be small enough for learners to begin practicing a simplified, but authentic, version of the whole task within the first few days. The final cluster must be large enough to rely on all of the constituent skills identified in the preliminary analysis. Specific problems should only be large enough to provide examples or practice of a specific skill (4C/ID).

Sequence

- Learning objects should be presented in order of increasing complexity, beginning with the epitome or simplest case (Elaboration Theory).
- Learning objects should be sequenced in an order that simulates the real-world performance with increasing fidelity. Because more than one learning object can be created from a single work model, and because the learning objects thus produced will function in instructionally equivalent ways, instructionally equivalent learning objects can be substituted for each other in the sequence (Work Model Synthesis).
- Learning objects should be sequenced according to their difficulty order on the unidimensional expertise scales. Because the difficulty of some learning objects may be indistinguishable from others (in terms of placement on the

scale), difficulty-equivalent objects can be substituted for each other in the sequence in a manner similar to computer adaptive testing (Domain Theory).

- Learning objects should be sequenced according to their level and type, and in order to promote transfer when possible. Macro-level skill clusters should be sequenced in a part-task manner, meaning that skills are taught one at a time and gradually combined. Meso-level case types should be sequenced according to a whole-task order, in which all skills are taught simultaneously. Micro-level specific problems can be sequenced in common simple-to-complex order or, when feasible, in a random sequence in order to promote transfer (4C/ID).

In Chapter 4 these principles will be combined with new work, following the theory-building methodology, to create a new instructional design theory. Finally, Chapter 5 will present a theory-testing methodology by which the design theory and validity argument using the evidence supporting the theory can be improved over time.

CHAPTER FOUR

LEARNING OBJECT DESIGN AND SEQUENCING THEORY

This chapter presents a new instructional design theory created specifically to support learning object design and sequencing called Learning Object Design and Sequencing Theory (hereafter LODAS, pronounced “lotus”). The chapter begins with a summary of the theory-building process and demonstration of the manner in which LODAS was developed according to the methodology presented in Chapter 2. The chapter then presents and exemplifies each component of LODAS in detail.

Applying the Theory-building Methodology

This section provides an overview of the manner in which the theory-building process outlined in Chapter 3 guided the completion of this study; the steps of that process are presented for reference during this discussion. Most of the steps are discussed individually, while some are discussed as a group.

> *1. Define the Purpose of the Theory*. Because of the technical nature of the learning object dialog in the field and the extremely narrow definition of "learning objects" proposed by the few existing instructional approaches that explicitly support learning objects, the purpose of the new theory is to provide an instructional view of learning objects, where "learning objects" are defined broadly enough to encompass resources currently existing on the Internet. This type of support is necessary to support the explicitly instructional use of learning objects (e.g., as opposed to their use in knowledge management) and facilitate a significant amount of reusability across objects.

2. Select a Paradigm for the Theory. The paradigm of LODAS is one intact model, meaning that the goals, values, conditions, and methods of the theory should be applied as a whole. For purposes of implementation, LODAS can be divided into two large sections: instruction design prescriptions and learning object design prescriptions. While LODAS is intended to be applied wholly as an intact model, a developer may choose to employ only one of these major sections. Any selected section should be implemented wholly as an intact model. Constraining the implementation of LODAS in this way allows the developer to predict the effectiveness with which the component methods will function. It also allows the learning object design prescriptions to be used separately, in conjunction with other instructional design theories. This is significant because it makes instructional design theory-general learning object design guidance available for use with any instructional approach.

3. Determine the Specific Domain, Situation, or Scope of the Theory. The scope of LODAS is the domain of complex cognitive problem solving, which van Merriënboer (1997) defines as "complex, in the sense that (1) they comprise a set of constituent skills, and (2) at least some of those skills involve conscious processing; and cognitive, indicating that the majority of constituent skills is [are] in the cognitive domain – as opposed to the affective or motor domain (p. 19)." Because of the close relationship between methods and conditionalities, and because the majority of LODAS' instructional methods are taken from the Four-Component Instructional Design Model, LODAS also exhibits this conditionality.

LODAS may be used in instructionally effective ways in other contexts; however, the rigor of the approach is probably overkill for non-complex problems.

4. Identify an Optimal Process on Which to Model the Theory. Identifying an optimal process on which to model LODAS was difficult, as no “optimal” process existed in the real world. This step reminds the theory-builder to use a grounded approach when such is available; however, while there are several existing approaches to the instructional use of learning objects (as reviewed in Chapters 1 and 3), none defines learning objects in a way broad enough to include the terabytes of resources already available online. The author considers this a significant weakness of other theories, as reusability should be the primary concern of object-oriented approaches and these theories ignore thousands of gigabytes of existing resources. Therefore, a strictly grounded theory approach was not usable for this study. Rather, an optimal process was conceptualized, rather than empirically identified, in which content would be broken down into smaller chunks, instructional design would be performed for the content chunks, the instructional design would be translated into learning object design, and the learning objects would be sequenced. Having identified the general process conceptually, several instructional design theories were reviewed in order to identify existing, proven methods that could be used to complete the optimal process. Where existing methods were not found, new work was done for this study (as in the case of the learning object taxonomy). Finally, existing methods and new work were synthesized into a single instructional design theory.

5. Develop general criteria for goals, methods, and conditions. In order to develop the general criteria, the author brainstormed the characteristics of successful theories. From consideration and elaboration of this list, and with the help of conceptual reviewers, the general criteria for judging goals, methods, and conditions were determined as: Significance, Communicability, Parsimony, Realizability, and Sustainability. These criteria relate to the goals, methods, and conditions in that the absence of any one of these characteristics from the goals, methods, or conditions would doom the theory to failure, or in other words, rob the theory of any utility. For example, if the methods are too difficult to understand (Communicability, Parsimony) they will never be properly applied; if the conditions are unrealistic (Realizability) application of the theory will never succeed; if the goals of the theory are trivial (Significance) no one will ever bother to implement the theory. These criteria and the degree to which LODAS meets them are described below in the section **Goals and Values** (p.52).

6. Develop Goals for the Theory, The specific goals of LODAS were developed through an in-depth examination of the purpose as stated above in (1). Reviews of literature, technical specifications, and discussions with experts in the field revealed the need for an instructional design theory that would

- catalyze dialog around the use of learning objects in an instructional design context,
- provide explicit support for the design of learning objects,
- provide explicit support for the sequencing of learning objects,
- provide learning object support in a reusable manner, and

- provide forward compatibility with expertise-based domain and learner modeling research.

These goals were selected as the purpose was compared against the criteria, and they are actually more specific sub-statements of the purpose. If the goals are accomplishable as written they would prove significant, particularly learning object design guidelines that are usable across instructional design theories. They have been written so as to be easily understandable (communicable). They have been chosen so as to be free of needless complexity (parsimony). The statements are made in a manner which is instructional design theory neutral and technology independent, so that they may be as broadly applicable and easily implementable as possible (realizability). Finally, they are grounded in current thinking regarding the future of the field of instructional design and are not tied to a specific technology (sustainability). The degree to which LODAS meets these goals is discussed in the following section, **Goals and Values** (p. 52).

7. Develop Methods for the Theory. The methods of LODAS were developed through a process of reviewing and synthesizing existing theories in order to flesh out the optimal process previously identified. Having identified the general process conceptually, several instructional design theories were reviewed in order to identify existing, proven methods that could be used to engage in the optimal process. Where existing methods were not found, new work was done for this study. Finally, new and existing methods were synthesized into a single instructional design theory. The criteria played a significant role in the selection and development of methods, as they did in the development of the goals. Each proposed method was compared to the optimal process, the current set of methods selected, and the criteria. If the method met a need in the optimal process as yet unfilled by the set of selected methods, and if the method met the criteria, it was selected for inclusion in the theory. Finally, where no method could be selected to plug in to the optimal process according to the rule described above, new methods were created according to the instructional deign theory need and the criteria.

8. Identify Conditions for the Theory. Because of the close relationship between methods and conditionalities, and because many of LODAS' methods are taken from existing instructional design theories, LODAS exhibits many of the conditionalities of its constituent theories. LODAS may be used in instructionally effective ways in environments that do not exhibit the proper conditions; however, there may be problems (e.g., the rigor of the approach may be overkill for non-complex problems). The criteria played the same role in the identification and expression of the conditions as with the goals and methods.

9. Create a Variable Taxonomy for the Theory. The instructional methods identified in existing theories and new methods created as part of this study were combined to create a taxonomy of methods. The taxonomy differentiates between four major types of methods, those that function to Analyze and Synthesize Content, Design Practice and Information Presentation, Select or Design Learning Objects, Design Learning Object Sequencing. An additional taxonomy of learning object types was created as part of LODAS, but this is not the taxonomy described in the Theory-building Methodology. Each of these methods is described in detail in the section **LODAS' Methods** (p. 58).

10. Finalize the Theory Prototype, 11. Formatively Research the Prototype Theory, 12. Revisit the Goals, Methods, and Conditions. The LODAS prototype was finalized in early drafts of this study and read by experts in the both the fields of instructional design and learning objects as a preliminary type of formative research. Dr. Charles Reigeluth, an expert in instructional design theory who has been quoted frequently in this study, read early drafts of chapters two, four, and five of this study. He summarized his detailed response to as follows.

> This is a very impressive dissertation. Wiley has done a fine job of building an instructional theory based on learning objects. Of course, as with any theory at this stage of development, there are questions about some substantive elements of the theory. For example, shouldn't there also be rules for sequencing the different instructional strategies (such as demonstrations and practice), in addition to the rules for sequencing the cases that go into the instruction? However, omissions of this nature are common in the early versions of a design theory. And the creative scholarly contribution represented in this dissertation is truly impressive.
>
> Naturally, this effort should be viewed as the first step in a lengthy process of testing and improving the theory (formative research), followed by some studies to prove the worth of the revised theory (summative research). I am concerned about the focus in Chapter 5 on "proving" the theory, because the major research concerns for design theory should be pursuing optimality and identifying conditionalities, given that many different methods may suffice to reach the goals, but some methods will

> inevitably be superior than others under different conditions. What is the point in "proving" that a given method of cutting the lawn will work? Cutting it with scissors will work. The issue should be to find the best known method for cutting the lawn under each of various conditions, and then to continually search for better methods. I see too much emphasis in Chapter 5 on "proving" the theory (which seems as ludicrous as proving that cutting a lawn with scissors works) and not enough emphasis on "improving" the theory—striving to enhance its optimality. While I recognize that many people do not recognize that research on design theory should focus on optimality rather than validity and on improving rather than on proving, I would expect to see such a recognition in scholarly work of this high caliber.
>
> This line of work appears very promising, and I look forward to seeing its progress. Congratulations to Wiley on cutting-edge work that holds great promise (Reigeluth, personal communication, June 16, 2000).

Reigeluth's comments regarding the theory's need for further research and improvement are in agreement with the author's feelings. His comments regarding theory-testing are also extremely thought provoking (as they raise interesting questions regarding the focus of theory-testing) and Reigeluth has expressed interest in a continuing discussion regarding theory-testing after this study is finalized.

Brandon Muramatsu is Project Director of the National Engineering Education Delivery System (NEEDS) and the Science, Mathematics, Engineering, and Technology Education (SMETE) digital libraries at the University of California at Berkeley. The NEEDS and SMETE projects are the largest NSF-funded digital libraries of learning objects in the country. Mr. Muramatsu is also actively involved in many of the learning object initiatives mentioned earlier, such as the IEEE Learning Technology Standards Committee and the IMS Project. In addition to providing detailed comments on LODAS, he summarized his response to chapter four as follows.

> The emergence of the World Wide Web and networked learning resources in the early 1990’s as a viable means of sharing and re-use of education materials fundamentally changed our view of the way education and learning can be delivered. We are seeing an enormous quantity of materials being developed to support education and learning However, locating these resources and evaluating their quality and applicability for learning has been quite difficult.
>
> For the last decade NEEDS—the National Engineering Education Delivery System has been developing a digital library of educational resources for engineering, and now science, mathematics and technology. We view locating (and cataloging) resources for re-use to be an evolution and education process. We have long recognized the need for many of the ideas expressed in the Learning Object Design and Sequencing theory and taxonomy to be applied to resources, but have been without a formalized expression of these ideas. It has taken an evolution and maturation in the practice of developing digital learning resources to set the stage for the

existence of LODAS. We have seen the development of computer-based instructional resources progress from a few experts developing materials to anyone, anywhere, anytime developing and using these materials. At the same time, we have seen the importance of metadata standards for learning objects grow dramatically in the last few years. These metadata standards have focused on the basic description of learning objects available in a networked environment.

The Learning Object Design and Sequencing theory and taxonomy takes the next step beyond metadata in facilitating the use and re-use of these networked resources. LODAS seems to be able to bridge the gap between learning object taxonomies and various instructional design theories in use today. It addresses the difficult problem of supporting the re-use of **existing** resources in an instructionally meaningful way. It provides an approachable methodology for instructors and instructional designers to use, which will hopefully facilitate its widespread acceptance.

It is important to recognize that the LODAS theory can be applied to existing resources, in the real world, today. The LODAS taxonomy is readily applicable to the resources indexed in NEEDS; it provides a formal description of many of the principles we have been using for the last decade. It also acknowledges the wide range of resources available; and appears to handle granularity from individual JAVA applets to full courseware applications. I look forward to working with David to integrate LODAS principles into NEEDS and follow-on National SMETE

> Digital Libraries, and expect our users will see an improvement in quality of service as a result (Muramatsu, personal communication, June 14, 2000).

Muramatsu's recognition of the real-world applicability of LODAS and desire to integrate LODAS principles into the Berkeley digital libraries, combined with funding from the National Science Foundation to carry out this work, will provide fertile ground for implementations of LODAS, and bodes well for its future.

This conceptual formative research in which expert reviewers provided detailed feedback on chapter drafts provided valuable feedback, not only for the instructional design theory itself, but for the theory-building process as well. Specific areas identified by reviewers were revisited and adjusted according to their comments and additional work, as seemed appropriate. Both the theory-building process and theory have been improved in this manner.

13. Plan for Theory-Testing. Chapter 2 presented a theory-testing rationale for testing and improving LODAS, stating that the development of a theory-testing plan should occur simultaneously with the theory-building activities. Chapter 5 presents an actual theory-testing plan developed specifically for LODAS, stressing Reigeluth and Frick's (1999) assertion that instructional design theories are never perfected and in constant need of improvement. The theory-testing framework has been designed specifically for testing and improving LODAS, but may also be useful in the formative research of other instructional design theories.

14. Write Up the Theory. In addition to publication in the dissertation, a book based on many of the ideas presented in LODAS has been accepted for publication by AECT. This exposure will provide the first large-scale opportunity for discussion and implementation of the theory, and should provide additional valuable feedback for improvement.

15. Implement and Improve the Theory. Finally, Chapters 4 provides a conceptual implementation of LODAS in the domain of Music Theory. This conceptual implementation revealed weaknesses in LODAS that have been strengthened prior to the finalization of this study. Formal implementation of the theory will first take place after the completion of this study, and should yield empirical data that will provide additional insights into opportunities for improving LODAS.

PRESENTATION OF THE THEORY

Like other instructional design theories, LODAS is comprised of goals, values, conditions, and methods. The remainder of this chapter will present and describe these goals, values, conditions, and methods in detail.

Goals and Values of LODAS

The development of LODAS has been motivated and directed by certain goals and values. The pervasive and influential nature of these underlying aims and assumptions requires that they be made explicit, so that the would-be user of the theory can judge whether these goals and values are compatible with their own ideals.

LODAS has been designed to meet four main goals.

1. Catalyze dialog around the use of learning objects in an instructional design context,
2. provide explicit support for the design of learning objects,
3. provide explicit support for the sequencing of learning objects, and

4. provide learning object support in a reusable manner, and
5. provide forward compatibility with expertise-based domain and learner modeling research.

Additionally, there are several values that guided both the creation and the write-up of the theory. These values speak to desired characteristics of the theory itself, and serve as criteria for each of the goals, conditions, and methods.

- Significance – Instantiation of LODAS should make instructionally possible either that which was not previously possible or degrees of precision, efficiency, etc., that were not previously possible.
- Communicability – LODAS must be amenable to formalization and explanation to such an extant that it can be adequately communicated to other individuals.
- Parsimony – LODAS should not be so complicated or time-consuming as to preclude its implementation.
- Realizability – LODAS must be implementable within the learning objects technology framework.
- Sustainability – LODAS must be a first step toward, and not be incompatible with, a long-term research and improvement agenda.

One goal or value that LODAS will not be able to fulfill directly, but nonetheless played a critical role in motivating and guiding the development of the theory, is that of increasing access to educational opportunity for those who to date have not had it. While the connection between this goal and LODAS as outlined may not be readily apparent, the implications of high-quality instructional components that can be developed once and used a potentially infinite number of times are great – the ratio of production cost to number of uses of the learning object approaches zero. It is the author's hope that instructional design theories such as LODAS will facilitate the provision of free or no cost instruction.

Conditions for Application of the Theory

Nelson (1998) opens her discussion of instructional design theory conditions with the following statement.

> Because not every instructional approach is effective in every learning context, it is necessary to determine *when* a particular approach might be the best possible match for the learner's needs, the instructor's teaching style, the learning environment, and the instructional goals. It is also important to determine *how* an instructional approach could be used in a given context. Does the whole approach need to be implemented to maintain its integrity, or can certain parts or strategies be used in isolation? This section will consider conditions related to when the [instructional design] theory is most appropriate and how it is best implemented (p. 85).

Many of the conditional constraints of LODAS are inherited from the instructional design theories from which methods have been borrowed; others come from its close ties to instructional technology, as explained below. This following discussion of conditions follows Nelson’s (1998) condition discussion format.

The close relationship between the content, learning environment, learners, and instructors is reflected in the constraints. Failure to meet the conditions outlined below will almost certainly create large amounts of variance in theory-testing activities.

Type of Content. The content which students engage is frequently divided into three types according to what the student is expected to gain from the instruction: knowledge, skills, and attitudes. The content analysis and synthesis procedures borrowed from Domain Theory and Work Model Synthesis both stress what could be called skill or "expertise representation" (Wiley, in press), in which an emphasis is placed on valuable performances which people perform in the real world. Additionally, the Four-Component Instructional Design (4C/ID) model, from which many of the instructional design strategies of the current theory are borrowed, states very clearly that it is targeted at the domain of *complex cognitive skills*. Van Merriënboer (1997) described complex cognitive skills as:

- complex, in the sense that (1) they comprise a set of constituent skills, and (2) at least some of those skills involve conscious processing; and
- cognitive, indicating that the majority of constituent skills is [are] in the cognitive domain – as opposed to the affective or motor domain (p. 19).

LODAS also exhibits this content restriction to the set of complex cognitive skills, like computer programming or air traffic control. While the methods described herein would most likely facilitate successful learning in any skill domain, the rigor of the approach may be overkill for non-complex skills that could be taught in less comprehensive or exhaustive ways.

Learning Environment. LODAS will function best in a learning environment in which instructor and learner feel a sense of joint responsibility toward learning, and assessment is appreciated as a progress-facilitating tool. Learners should take a keen interest in their progress through the domain of expertise, participating willingly in formative assessments and using the results to monitor their own learning. Instructors should use assessments formatively, and be willing to support learners' efforts to interpret and use the results.

Learner Characteristics. Learners who can and will monitor and regulate their own learning have a greater chance of succeeding in learning environments built around LODAS than those that cannot or will not because LODAS provides students with access to real-time information regarding their current level of expertise. Accordingly, a change in thinking about assessment results will be necessary for learners. Not only do learners need to be willing to participate in formative assessments without fear of "failure," they need to learn that lower scores do not always have negative consequences. They need to learn how to read and interpret domain maps, their current positions on the maps, and learn to see low performance as potential for improvement instead of a summative judgment. While coaching will be necessary to orient learners to a new way of thinking about assessments, the learning environment, and the maps, learners who are willing to accept this paradigm of assessment and self-control have a greater chance of succeeding in the environment. If students are unwilling to accept and work in this paradigm, students will likely struggle to succeed in LODAS-based learning environments.

Additionally, learners must have some competence (or a willingness to gain it) in computer use in order to use the learning environment. Again, some training may be necessary to help students gain a level of competence that will facilitate successful learning in the technology-enhanced environment. Insufficient ability to use computers will also be likely to cause students to struggle in LODAS-based learning environments.

Instructor Characteristics. Instructors must be willing to empower students in their own learning, i.e., they must be willing to relinquish some control of the learning process. Instructors must be willing to use formative assessments as diagnostic tools as opposed to tools for sorting students. They must be willing to explain the results of formative assessments to students without being judgmental. Instructors must be willing to take the role of “the guide on the side.”

Instructors will also need computer competency to succeed in using the new theory. Even if they employ learning objects that have been designed and sequenced by a third party, instructors must be at least capable of using and understanding the environment as a student.

Implementing LODAS. For purposes of implementation, LODAS can be divided into two large sections: instruction design prescriptions and learning object design prescriptions. While it is intended that LODAS be applied wholly as an intact model, a developer may choose to employ only one of these major sections. Either of these sections should be implemented as an intact model. While it may be possible to isolate and apply some of the individual methods within these sections outside of the overall theoretical context, no predictions can be made as to the manner in which the individual methods will function.

LODAS will also work most efficiently when employed in conjunction with a digital library of learning objects. While LODAS will allow the designer to create every object they use by hand, one of LODAS’ goals is to promote reuse of objects, and its implementation will be most efficient when objects are being reused.

While LODAS has been designed around a definition of learning object that implies computer-based learning environments, it may be possible to use the theory to create a paper-based curriculum. This situation is undesirable, because the potential reusability of discrete, physical learning objects is substantially lower than that of the digital learning object.

LODAS' METHODS

LODAS describes a process for transforming an undifferentiated content domain into specifications for (1) the scope and design of learning objects, and (2) the sequencing or combination of learning objects. The methods fall into six categories, Preliminary activities, Content analysis and synthesis, Practice and information presentation design, Learning object selection or design, Learning object sequencing, and Loop back for quality improvement. As these methods are described below a conceptual implementation of LODAS will be described in parallel, in which the abstract methods are demonstrated in the context of a first semester undergraduate Music Theory course. This example is by no means perfect, and no doubt an actual (rather than conceptual) application of the theory would improve it. The example should be taken as exemplifying the methods and not an attempt to make a contribution to understanding the domain Music Theory.

Preliminary Activities

Determine appropriateness. Before employing LODAS, instructors or instructional designers should compare the goals, values, and conditions identified above with their own values and environments. As stated previously, the methods identified in LODAS will not always be the optimal instruction methods. A decision to either commit to the approach or take some other approach should be made before moving any further into the process.

Example. Cheryl Richardson, the instructor of a freshman music theory course, desires that her students be able to analyze pieces of music by the end of her course. Having heard about LODAS at a conference, Cheryl picks up a book and examines the theory, comparing it with her own attitude about teaching and her goals for her students. She decides to give LODAS a try.

Analyze and Synthesize Content

Overview. This group of methods functions together to transform an undifferentiated content domain into specifications for the content scope and sequencing of learning objects.

Principled skill decomposition (van Merriënboer, 1997). In this step, the complex cognitive skill to be taught is broken down into its constituent skills. Principled skill decomposition, task analysis (Merrill, 1976), or another procedure is used to break the complex skill down into its component parts, as demonstrated in Figure 1. It is up to the designer to decide when the skills are at a low enough level of granularity, and to stop the analysis.

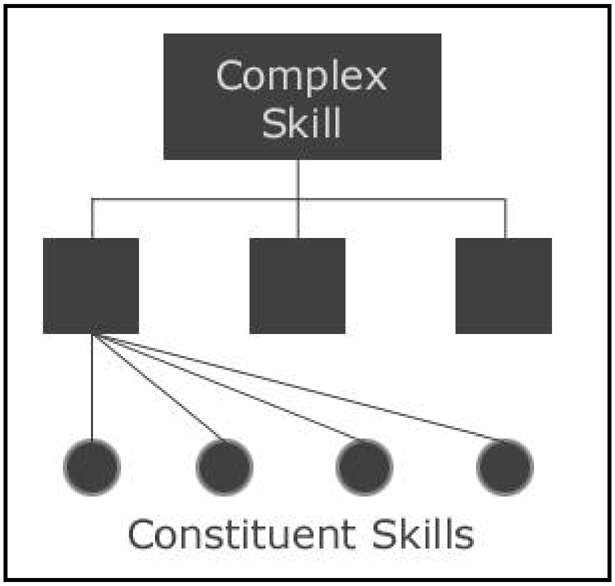


Figure 1. Principled Skill Decomposition. A complex skill is broken down into constituent skills through principled skill decomposition.

Example. Cheryl begins with the complex cognitive skill “analyze a piece of music.” She decomposes the skill into constituent skills such as “identify intervals,” “stack notes in thirds,” and “identify non-chord tones.”

Synthesize work models (Gibbons, et al., 1995). Gibbons described Work Model Synthesis as “systematically combin[ing] and recombin[ing] tasks and objectives that through task analysis procedures have been fragmented at a low level.” Thus, work models are collections of individual objectives, or in the case of LODAS, constituent skills, that have been recombined into activities that real people perform in the real world, and therefore have value in the real world. After performing principled skill decomposition, the constituent skills are synthesized (recombined back) into work models, as shown in Figure 2.

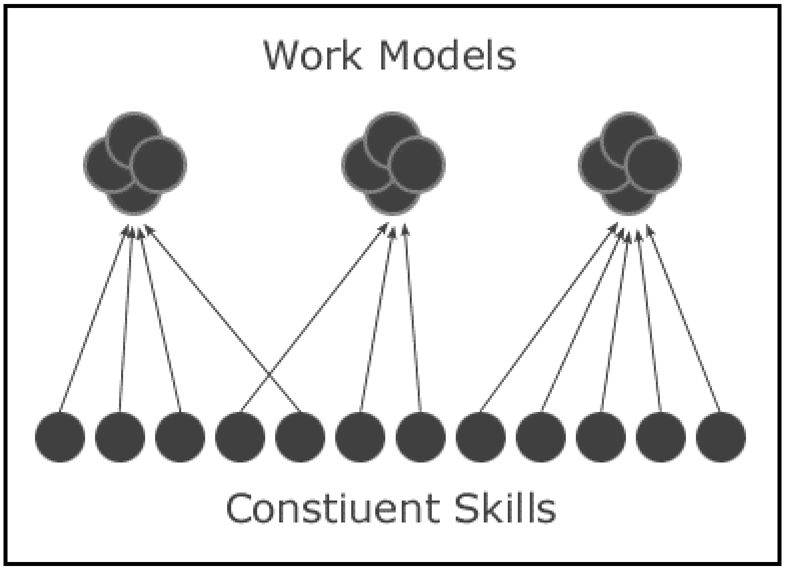


Figure 2. Work Model Synthesis. Work models are created through the synthesis of constituent skills.

Example. Cheryl chooses to synthesize the constituent skills “identify intervals” and “stack notes in thirds” into a work model called “Identify the root, quality, and inversion of chords.” The identification of chords is a valuable skill that composers, arrangers, and musicians who frequently improvise (such as jazz musicians) perform in the real world. Chord identification also plays an important role in music performance, as coaches or conductors will give instruction such as “always sing on the high side of the third.” Chord identification will likely hold more relevance for learners than the individual constituent skills combined to comprise the work model (who would want to stack notes in thirds?). The resulting work model can be used as a template or specification for a number of instructionally equivalent learning objects, as well as multiple equivalent assessments like those found in an item bank.

Identify the dimensionality of the domain (Bunderson, Newby, and Wiley, 2000). By placing itself within the realm of cognitive skills, the new instructional design theory avoids a major pitfall associated with some previous attempts at computer-assisted instruction: the difficulty of knowledge representation. Because LODAS models what a learner can *do* with their knowledge (cognitive skills), and does not attempt to independently represent in machine form what they *know*, the problem of representing knowledge is sidestepped. The result can be called "expertise representation" (Wiley, 2000).

Every content domain has dimensions of expertise. These dimensions may be regarded as the major types of activities in which experts in the domain engage. In the domain of language learning, these dimensions might be reading, writing, speaking and listening. The process of discovering these dimensions can include qualitative methods (such as review and synthesis of existing literature or interviews with SMEs) and quantitative methods (such as factor analysis or smallest space analysis).

Example. Taking into account her subject matter knowledge, pertinent literature, and through talking with peers, Cheryl postulates that the domain of music analysis has two primary dimensions of expertise: harmonic analysis and formal analysis. She realizes that she may be wrong in the number or identification of the dimensions, but feels confident that these will serve as a useful starting point for the organization of her instruction.

<u>Place work models on scales</u>. Once work models have been synthesized and dimensions of expertise have been identified, the dimensions can be thought about in a new manner. The dimensions of expertise can be considered scales in the sense of number scales, the implication being that some quantity is increasing the further one travels from the scales' point of origin. Assuming that the quantities represented by

distance along the scale are simultaneously difficulty of tasks and expertise of learners, work models can be anchored or positioned on the scales according to the expertise needed to successfully complete them, or in other words, the work models can be ordered along the scales according to their difficulty. This concept is illustrated in Figure 3.

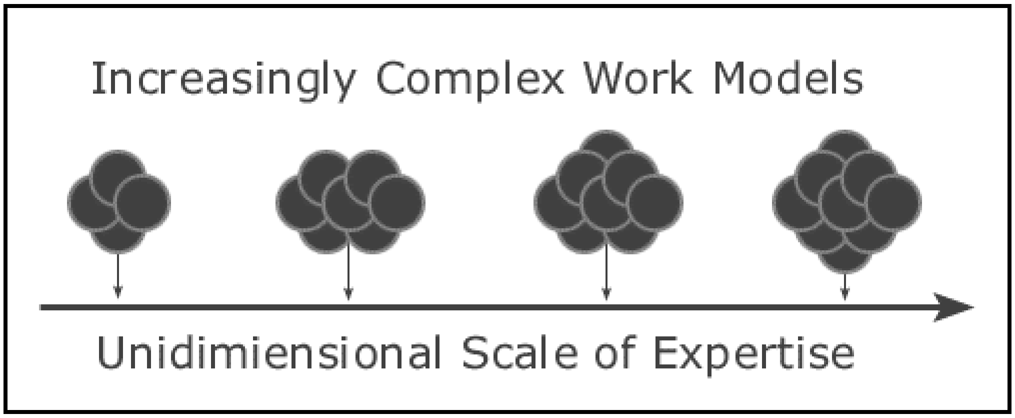

Increasingly Complex Work Models
Unidimiensional Scale of Expertise

Figure 3. A Unidimensional Scale of Expertise. Work models should be placed on the scales of expertise in order of increasing complexity.

At this point in the analysis and synthesis process, the instructional designer must make two judgments. First, which scale does each work model belong to? Second, what is the relative difficulty of work models belonging to each scale? While these work model placements should be made according to the best information (existing research, etc.) available, it is fully expected that the positions will evolve with time and greater understanding of the domain.

Example. Cheryl now examines the several work models synthesized previously and first tries to determine which scale they should be placed on. She associates the work models “Identify the root, quality, and inversion of chords,” “Identify non-chord tones,” and “Identify the harmonic function of chords” with the Harmonic Analysis scale. She groups the work models “Identify cadences,” “Identify motives and phrases,” and “Identify periods” with the Formal Analysis scale.

Next, Cheryl must decide on the relative difficulty of the work models associated with each scale. For the Formal Analysis scale the task is straightforward, as there is a strong prerequisite relationship between the work models. She orders the work models as follows: “Identify cadences,” “Identify motives and phrases,” and “Identify periods.” The problem is not so straightforward with Harmonic Analysis, however. While both “Identify non-chord tones” and “Identify the harmonic function of chords” depend on the ability to identify chords, there is no clear prerequisite relationship between them, and Cheryl is unable to make a principled decision regarding their relative difficulty other than estimating them to be equally difficult. She realizes that this is fine; the data will tell her more, later.

At this point, Cheryl has a domain map that looks similar to Figure 4.

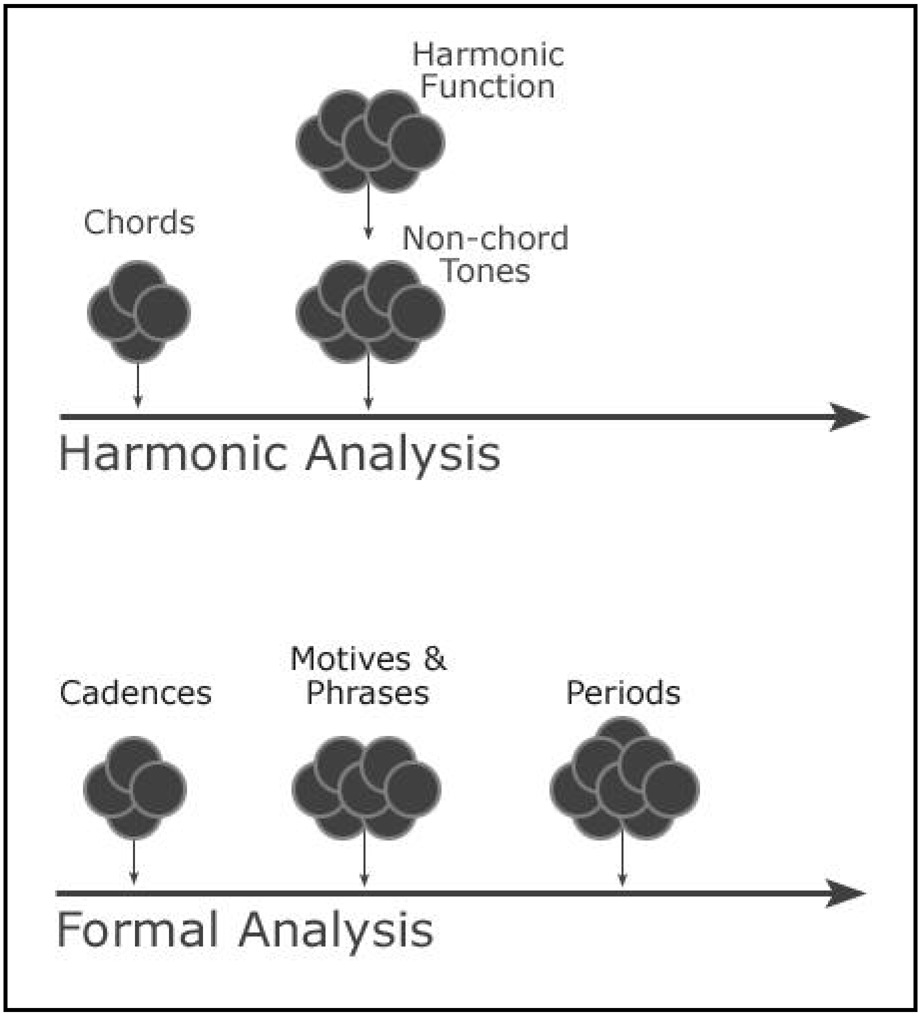


Figure 4. Domain Map. Cheryl's first draft of a domain map for music analysis. The two horizontal arrows represent the two scales Cheryl identified. The point of origin for each scale is its left end. Work models are represented as becoming increasingly difficult or complex by their size increasing as they get farther from the point of origin. Arrows from work models to the scale indicate the work models' position on the scale. The exact distance between two work models is unimportant at this stage of development, only the left to right qualitative ordering. The work models Harmonic Function and Non-chord Tones are estimated to have similar difficulty.

Synthesize integrated work models. Each of the scales or dimensions of expertise can be divided into sections roughly corresponding to broad levels of expertise, such as novice, intermediate, advanced, and expert. The individual work models placed at each level can be further synthesized into work models that integrate skills across dimensions of expertise. If it is possible to create a work model that includes all work models at a given level of expertise on each scale, this integrated work model would certainly be equivalent to the instructional epitome of the Simplifying Conditions Method of Elaboration Theory (Reigeluth, 1999). However, in many cases this may be impossible early on in the learning of complex tasks (van Merriënboer, 1997). For example, in the language learning domain, work models from reading and writing may be easily combined at the novice level, but an authentic task that includes novice-level work models from reading, writing, listening, and speaking may not be possible to construct, as novice level performance generally lacks such integration. However, progressing up the scale of expertise, such integrated work models become easier to construct. The final integrated work model, which combines expert-level performance on work models from all scales, *is the complex cognitive skill* originally identified for instruction, as illustrated in Figure 4.

Example. At this point Cheryl thanks her lucky stars: her domain is fairly simple, with only two scales (or dimensions) of expertise and a small number of work models. Creating integrated work models should be possible even at lower levels of expertise. Cheryl combines the easiest work models on each scale, chord identification and cadence identification, into the integrated work model “Classify cadence types.” (While a cadence is loosely defined as a “resting place,” there are a number of cadence types that are classified according to their composite chords.) She then combines the non-chord tone identification and motive and phrase identification work models into the integrated work model “Reduce Phrases.” (Phrase reduction consists of removing embellishments and non-chord tones from phrases). Finally, Cheryl combines the harmonic function identification and period identification work models into the integrated work model, “Analyze a piece of music.” (Cheryl feels that when integrated, these structural and harmonic techniques comprise a sufficient level of analysis expertise for first semester students.)

Expose domain map to expert review. At this point a fairly sophisticated map of the domain of interest exists, as the type of map represented in Figure 4 is now expanded through the addition of integrated work models. Whether or not they have been consulted previously, domain or subject matter experts should review the map before it is used in design activities, as minor changes, or a loop back to the beginning of the process, may be necessary. This domain map is the central element in the creation of a blueprint for the instructional design.

Example. Cheryl sends her domain map to her friends Ed and Janice who teach similar courses at other universities. They agree that the map looks “pretty good,” and express excitement to see if the data reinforce her hypotheses or suggest others.

Design Practice and Information Presentation

Overview. These methods function together to specify the specific problems and instruction which will be instantiated in learning objects, based on the scope and sequence information resulting from the previous group of methods.

Classify the work models and constituent skills (van Merriënboer, 1997). Components of the complete complex cognitive skill (i.e., the constituent skills and work models) can be categorized into one of two groups: recurrent skills and nonrecurrent skills. Recurrent skills are those that are executed in the same manner each time (algorithmic methods). Nonrecurrent skills are those that differ widely according to the situation in which they are performed (heuristic skills). Each constituent skill or work model will additionally require some form of prior information in order to be completed successfully, whether in the form of prerequisite knowledge (such as procedures, rules, or concepts) for recurrent skills, or supportive knowledge (such as causal, conceptual, or mental models) for nonrecurrent skills. In other words, prerequisite knowledge assists in the completion of recurrent skills, and supportive knowledge supports the completion of nonrecurrent skills. Each of the constituent skills and work models should be classified as either recurrent or nonrecurrent, and necessary prerequisite or supportive knowledge should be identified. This identification can occur through the hierarchical analysis of facts, concepts, plans, and principles for prerequisite knowledge, and through an associative process that identifies relationships between chunks of information that facilitate the completion of nonrecurrent skills (van Merriënboer, 1997).

Example. Cheryl sits down again to review her work models. In thinking through several examples of each work model performance in order to make an initial recurrent / nonrecurrent classification for each of the work models, she notices an interesting trend: work models associated with harmonic analysis tend to be algorithmic, while those associated with formal analysis seem to be more heuristic. This is exactly the opposite of how she imagined things might be. Realizing that in later courses students will learn that special cases and exceptions turn (almost) all of music analysis into an exercise in heuristics, she finalizes her classification: chord, non-chord tone, and harmonic function work models are labeled recurrent, and cadences, motives and phrases, and periods are labeled nonrecurrent.

In order to identify prerequisites for each of the work models, Cheryl goes first to the constituent skills combined to form the work models. Taking these as a first level of prerequisite information and supportive knowledge, she analyzes each one level further down, at which point she begins reaching prerequisites for enrollment in her course, such as being able to "name notes on both bass and treble staffs."

Design practice and information presentation (van Merriënboer 1997). The 4C/ID model provides very specific prescriptions for the instruction of these four types of content: recurrent skills, prerequisite information, nonrecurrent skills, and supporting information. These methods are described in detail in van Merriënboer's book (1997), and are summarized here.

Whole-task practice. The emphasis of this theory is on *doing* rather than just *knowing*, and practice is at the heart of the instructional prescriptions. "Whole-task practice" is prescribed for nonrecurrent skills and provides the framework for the rest of the instructional design. Whole tasks, or "skill clusters" as van Merriënboer calls them, are similar to work models in that they represent an entire task. Skill clusters differ from work models in that work models represent a conscious reaction against over-fragmentation and therefore focus on performances perceived as valuable and relevant.

Another significant difference between LODAS and van Merriënboer's approach is the explicit multi-dimensionality of LODAS's domain map, whereas the 4C/ID model is not clear about its assumptions of dimensionality. In other words, if Cheryl were using van Merriënboer's approach, she would only have one dimension of expertise: music analysis. Bunderson, Newby, and Wiley (2000) believe that the incorrect assumption of unidimensionality of domain expertise (in other words, assuming that there is only one dimension of expertise when there are in fact several) will result in problems in the simple to complex ordering of work models. An example of one such problem is non-transitivity of work model difficulty, in which work model A is more difficult than work model B, work model B is more difficult than work model C, and work model C is somehow more difficult than A.

"The key process to be promoted [in whole-task practice] is *inductive processing*, which is the abstraction of schemata on the basis of concrete experience" (p. 174). Accordingly, the instructional design emphasis is on finding or creating instructional problems and worked examples for the whole task or work model of interest. This is accomplished through the creation of *case types*, which are several (simple to complex) versions of the whole task represented by the skill cluster. The case types are designed so that each case represents the entire to-be-learned work model in increasing degrees of difficulty, in a manner similar to the Simplifying Conditions Method of Elaboration Theory (Reigeluth, 1999). Case types can be used as abstractions of the classes of problems or worked examples that the learner will experience as they move through the skill cluster. Finally, the actual *specific problems*, or instances of the case types, are designed. Work models, case types, and specific problems are illustrated in Figure 5.

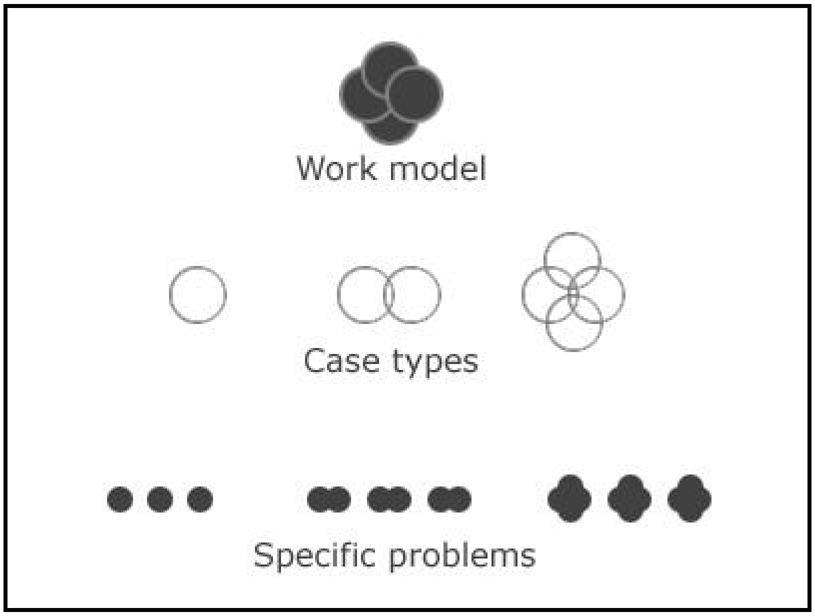


Figure 5. Work model, Case Types, and Specific Problems. Case types are practice specifications based on simple to complex versions of the work model. Specific problems are instantiations of the case types.

van Merriënboer recommends that case types within whole-task practice be sequenced from simple to more complex cases. However, specific problems within each case type should be ordered randomly to promote induction and transfer, to guard against tendencies to sequence problems in patterns. For example, in a set of interval identification items, one would prefer a random sequence of 12 items to a sequence of four items whose interval was a P4, four items whose interval was a P5, and four items whose interval was a tritone.

Part-task practice. Occasionally whole-task practice will provide enough practice opportunity with recurrent skills so that additional, explicit recurrent skills training may not be necessary. If it is necessary, the prescription of part-task practice for some recurrent skills is based on the Component Fluency Hypothesis (Carlson, Khoo, and

Elliot, 1990), which states that whole-task practice must occasionally be supplemented by targeted part-task practice in order to achieve the desired level of expertise in the performance of the whole task. Part-task practice should be considered if any of the following conditions hold for either the constituent skills identified during the principled skill decomposition or the work models synthesized afterwards.

- The skill or work model is associated with high cognitive load,
- the skill has vertical relationships in the constituent skills hierarchy, indicating that this skill enables the performance of many others,
- the skill maps into work models low on the scale of expertise, indicating that this skill enables the performance of many others, or
- the work model occurs low on the scale of expertise, indicating that this work model is prerequisite to the performance of others.

Part-task practice should not be designed simply because one or more of these conditions hold; however, it should be *considered* when this is the case. Automaticity is the goal of part-task practice, and drill and review techniques are the major form of practice. Van Merriënboer recommends that part-task practice be introduced in the instructional sequence when it is needed, or in other words, in the context of the work model in which the to-be-learned recurrent skill will actually be applied.

Just-in-time information presentation. “The design of information presentation is subordinate to, although integrated with, the design of practice” (van Merriënboer, 1997; p.170). Just-in-time (JIT) information presentation, or just-in-time instruction, is prescribed for prerequisite knowledge (the knowledge that supports the performance of recurrent skills). Just-in-time presentation can take one of two forms.

- Direct presentation in combination with the problem for which the instruction is relevant.
- Making the information easily accessible in job aids during practice.

  When the instructor or instructional designer has control over the learning environment, presentation in the problem context is the preferred method of JIT instruction. However, when learning will be self-instructional or in other scenarios over which the designer has little control, job aids are preferred. When JIT presentation is considered as a scaffolding technique, the progression from integrated presentation, to job aids, to no support can be used as a fading technique (van Merriënboer, 1997).

Promoting elaboration and understanding. Instruction that facilitates elaboration is prescribed for supporting information (the information required to successfully complete nonrecurrent skills) so that the conceptual, causal, and other mental models that support successful completion of nonrecurrent skills can be present in the learners mind during practice. This information is presented either before (deductive) or after (inductive) case studies and examples, but before practice. The presentation should stress significant relationships and link into familiar knowledge. Taken together with expository and inquisitory approaches to presentation, these can be combined into the four main strategies for elaborative information presentation: deductive-inquisitory, deductive expository, inductive-inquisitory, and inductive-expository. The aim of elaborative information presentation is to provide learners with the mental models and heuristics they need to successfully engage in the performance of nonrecurrent skills.

Summary. This section has summarized the Four-Component Instructional Design (4C/ID) model. The instructional design of both instruction and practice can be completed as per the 4C/ID strategies. Once this design has been completed, it is ready to be expressed in learning objects.

Example. Cheryl begins by designing the practice and instruction for the work model "Identify the root, quality, and inversion of chords." This is a recurrent skill, and while she knows she does not necessarily need to use part-task drill and review style instructional techniques simply because the skill is classified recurrent, so many other work models depend on the students' ability to successfully identify chords that she decides to aim for automaticity and use drill and review techniques to help students overlearn the constituent skills of chord identification: identifying major and minor thirds (intervals) and stacking notes in thirds. Cheryl plans to use whole-task practice to teach chord analysis itself once these skills are overlearned, and will include part-task practice during the whole-task sequence when it is appropriate.

Cheryl must now design case types for her part-task and whole-task practice and worked examples and supporting instruction. She decides on the following.

Identify major and minor thirds (recurrent / part-task)

- Worked example case types: note pairs on the treble clef properly labeled M3 (major third) or m3 (minor third)
- Practice case types: unlabeled note pairs on the treble and bass clefs
- JIT instruction: Definition of M3 (4 half steps), definition of m3 (3 half steps), an explanation of M3 and m3 from a scale perspective.

Stack notes in thirds (recurrent / part-task)

- Worked example case types: triads already stacked in thirds (in root position), triads in first inversion presented with the same triads stacked in thirds, triads in second inversion presented with the same triads stacked in thirds, four notes (hymn or chorale examples) presented together with three notes stacked in thirds (these will all take place in the treble clef)
- Practice case types: triads already stacked in thirds (in root position), triads in first inversion presented with the same triads stacked in thirds, triads in second inversion presented with the same triads stacked in thirds, four notes (hymn or chorale examples) presented together with three notes stacked in thirds (these will take place in both the treble and bass clef)
- JIT Instruction: Procedure for stacking three notes in thirds (job aid), procedure for stacking four notes in thirds (job aid)

Identify chord root, inversion and quality (recurrent / part-task)

- Worked example case types: major chords in root position with the root labeled, major chords in first inversion with the root labeled, major chords in second inversion with the root labeled, minor chords in root position with the root labeled, minor chords in first inversion with the root labeled, minor chords in second inversion with the root labeled (these will all take place in the treble clef)
- Practice case types: major chords in root position with the root labeled, major chords in first inversion with the root labeled, major chords in second inversion with the root labeled, minor chords in root position with the root labeled, minor chords in first inversion with the root labeled, minor chords in

second inversion with the root labeled (these will take place in both the treble and bass clef)

- JIT Instruction: Procedure for identifying the root of the chord (job aid), procedure for identifying the inversion of the chord (job aid), procedure for identifying the quality of the chord (job aid)

Having designed case types for her worked examples and practice, and having specifically identified the instruction and job aids necessary to understand the examples and complete the practice, Cheryl decides on parameters for the examples and practice items. Note pairs for interval examples and practice will begin on notes e through c for the treble clef and g through e for the bass clef. The chords presented for rearrangement in the stacking examples and practice items will be c, f, and g. The chords presented for identification in the identify chords examples and practice will be c, f, g, d, and a.

Content that the design of the “Identify the root, quality, and inversion of chords” work model is sufficient, Cheryl continues on to design the practice and instruction for her other work models.

Select and / or Design Learning Objects

Overview. This group of methods functions to link specific problems, worked examples, and instruction designed previously to specific types of learning objects and provides guidance for the design of each learning object type.

Taxonomy of learning object types. All learning objects have certain qualities. It is the difference in the degree to which or manner in which they exhibit these qualities that makes one type of learning object different from another. This section introduces a taxonomy of learning object types with which the designer should familiarize herself.

This section is included as reference, and does not contain any design prescriptions. Design prescriptions based on the information included in the taxonomy and discussion are included in the following sections.

This taxonomy identifies five types of learning objects. Examples of these five object types are given below, followed by the taxonomy, which explicates their differences and similarities.

- *Single-type* - For example, a JPEG of a hand playing a chord on a piano keyboard.
- *Combined-intact* - For example, a video of a hand playing an arpeggiated chord on a piano keyboard with accompanying audio.
- *Combined-modifiable* - For example, a web page dynamically combining the previously mentioned JPEG and QuickTime file together with textual material, on-the-fly.
- *Generative-presentation* - For example, a JAVA applet capable of graphically generating a set of staff, clef, and notes and then positioning them appropriately to present a chord identification problem.
- *Generative-instructional* - For example, an EXECUTE instructional transaction shell (Merrill, 1999), which both instructs and provides practice for procedures, for example, the process of chord root, quality, and inversion identification.

Distinguishing between the learning object types is a matter of identifying the manner in which the object to be classified exhibits certain characteristics. These characteristics are critical attributes and are stable across environmentally disparate instances (e.g., the properties remain the same whether or not a digital library of learning objects exists or not).

Table 1 presents the taxonomy previously mentioned. The purpose of the taxonomy is to differentiate possible types of learning objects available to designers for use in instructional design. This taxonomy is not exhaustive in that it includes only learning object types that facilitate high degrees of reuse. Other types of learning objects that hamper or practically prevent reuse, (e.g., an entire digital textbook created in a format that prevents any of the individual media from being reused outside of the textbook context), have been purposefully excluded in order to discourage their creation. The taxonomy's characteristics' values (such as High, Medium, and Low) are purposefully fuzzy, as the purpose of this taxonomy is solely to facilitate inter-object comparison (i.e., it is *norm referenced*), and not to provide independent metrics for classifying learning objects out of context, such as file size in kilobytes (i.e., it is not meant to be *criterion referenced*). Table 1 is followed by a more in depth discussion of each of the characteristics or critical attributes of learning objects and the learning object types themselves.

Table 1. Taxonomy of Learning Object Types.

Learning Object Characteristic	**Single-type Learning Object**	**Combined-intact Learning Object**	**Combined-modifiable Learning Object**	**Generative-presentation Learning Object**	**Generative-instructional Learning Object**
Number of elements combined	One	Few	Many	Few - Many	Few - Many
Type of objects contained	Single	Single, Combined-intact	All	Single, Combined-intact	Single, Combined-intact, Generative-pres
Reusability of component objects	(not applicable)	Low	High	High	High
Common function	Exhibit, display	Pre-designed instruction or practice	Pre-designed instruction and / or practice	Exhibit, display	Computer-generated instruction and / or practice
Extra-object dependence	No	No	Yes	Yes / No	Yes
Type of logic contained in object	(not applicable)	None, or answer sheet based item scoring	None, or domain-specific instructional and assessment strategies	Domain-specific presentation strategies	Domain-independent presentation, instructional, and assessment strategies
Potential for inter-contextual reuse	High	Medium	Low	Low	High
Potential for intra-contextual reuse	Low	Low	Medium	High	High

Learning object characteristics. The characteristics in Table 1 are described in more detail below.

- *Number of elements combined* – Describes the number of individual elements (such as video clips, images, etc.) combined in order to make the learning object.
- *Type of objects contained* – Describes the type of objects that may be combined to form a new learning object.
- *Reusability of component objects*– Describes the degree of ease with which constituent objects may be individually accessed and reused.
- *Common function* – Describes the manner in which the object is generally used.
- *Extra-object dependence* – Describes whether the object needs information (such as location on the network) about learning objects other than itself.
- *Type of logic contained in object*– Describes the function of algorithms and procedures within the object.
- *Potential for inter-contextual reuse* – Describes the number of different instructional contexts in which the learning object may be used, that is, the object's potential for reuse in different content areas or domains.
- *Potential for intra-contextual reuse* – Describes the number of times the object may be used within the same content area or domain.

Learning object type definitions. The five types of learning objects have been exemplified and their characteristics have been described. While the creation of strict definitions for these types is a career's work in progress (as with the definition of learning object itself), the author's current best thinking with regard to definitions of each type is captured below.

- *Single-type* – An individual digital resource uncombined with any other, the Single-type learning object is generally a visual (or other) aid that serves an exhibit or example function.
- *Combined-intact* – A small number of digital resources combined at design time by the object's creator, whose constituent learning objects are not individually accessible (recoverable) from the Combined-intact object itself. The Combined-intact learning object may contain limited logic (e.g., the ability to perform answer sheet referenced item scoring) but should not contain complex internal logic (e.g., the capacity to independently grade a set of item forms or case types). Combined-intact learning objects should be single purpose, that is, they should provide either instruction or practice.
- *Combined-modifiable* – A larger number of digital resources combined by a computer in real-time when a request for the object is made, whose constituent learning objects are directly accessible (recoverable) from the Combined-modifiable object. Combined-modifiable learning objects frequently combine related instructional and practice-providing Combined-intact and Single-type objects in order to create a complete instructional sequence.
- *Generative-presentation* – Logic and structure for combining or generating and combining lower-level learning objects (Single-type and Combined-intact types). (such as "identify chords"). Generative-presentation learning objects

can either draw on network-accessible objects and combine them appropriately or generate (e.g., draw) objects and combine them to create presentations for use in instruction, practice, and testing. (Generative-presentation learning objects must be able to pass messages to other objects with assessment logic when used in practice or testing). While Generative-presentation learning objects have high intra-contextual reusability (they can be used over and over again in similar contexts), they have relatively low inter-contextual reusability (use in domains other than that for which they were designed).

- *Generative-instructional* – Logic and structure for combining learning objects (Single-type and Combined-intact types) and evaluating student interactions with those combinations, created to support the instantiation of abstract instructional strategies (such as "remember and perform a series of steps"). The transaction shells of Merrill's Instructional Transaction Theory (Merrill, 1999) would be classified as Generative-instructional learning objects. The Generative-instructional learning object is high in both intra-contextual and inter-contextual reusability.

Select learning object types and design learning objects. Before instantiating instructional design in learning objects, a decision must be made regarding which types of objects to use. Guidelines for selecting a learning object type based on the work model classification (as identified in **Design Practice and Information Presentation** above) follows. In other words, these guidelines are a bridge for the designer to follow from instructional design to learning object design. Van Merriënboer's distinction between recurrent and non-recurrent types of skills is closely paralleled by the learning object taxonomy's distinction between generative and combined learning objects. Instruction for highly recurrent (generative) domains may contain mostly generative objects, while instruction for highly nonrecurrent (heuristic) domains may contain mostly combined objects. As a general rule, use generative learning objects whenever possible; otherwise use combined objects.

1. To the degree allowed by development constraints (cost, time, etc.), each individual graphic, video, audio, textual, animation, and software element designed for use in the computer-based learning environment should be designed as a Single-type learning object, archived, and indexed with metadata. Each Combined or Generative object that utilizes Single-types should contain a list of references to it's constituent objects (this only needs to be done explicitly for Combined-intact, as the other objects will already contain references to their constituent objects). This avoids the cost of fully indexing each Single-type object while still providing a means of locating and reusing the object later.
2. References, job aids, and other prerequisite or supportive information should be designed as Generative-instructional, Single-type, or Combined-intact objects (in this order of preference). These objects can then be presented in worked examples, practice items, or independently.
3. If possible (i.e., if the skills are algorithmic), select or design Generative-instructional learning objects for each case type. Otherwise, design Combined-modifiable templates for each case type. These objects can be

populated (by the machine in the first case, and the designer in the second) with other objects to facilitate the efficient creation of specific problems for the case type.

4. If Generative-instructional learning objects were created previously, specific problems will be created by the computer. Otherwise, specific problems should be designed as Combined-modifiable objects, so that the objects previously designed for information presentation and the problem can be combined dynamically at runtime into the Combined-modifiable specific problem object. This allows the designer to add adaptive features to the learning environment in the future, through the development of combination rules based on individual differences, for example. This also allows the inclusion of problem-specific information in the problem without diminishing the reusability of the component learning objects.
5. Any other objects needed to populate the Generative-instructional learning objects or case type templates should be designed as Generative-presentation, Single-type objects, or Combined-intact objects (in this order of preference). These objects can then be presented in the context of multiple problems, worked examples, and instruction.
6. Design Combined-modifiable learning objects for each entire work model. Using Combined-modifiable objects for work models allows the designer to (1) design *specific problems* and their components as independent learning objects that can be reused in other work models, and (2) combine and sequence the work models themselves at runtime.

7. Design Generative-instructional or Combined-modifiable learning objects for the integrative work models. Some Generative-presentation, Combined-intact, and Single-type objects previously developed can be reused to populate this object. Necessary objects that do not yet exist should be created as per the principles above, as Generative-presentation, Single-type, or Combined-intact learning objects.

Taking these guidelines into account, identify the number and type of learning objects required for each of the work models on each of the scales. Next, identify places in which a learning object may be reused across problems or work models. If a digital library populated with learning objects exists, select existing learning objects appropriate for use in the current work models. Finally, design the remaining learning objects based on the instructional design already completed.

Once a decision has been made regarding what learning object types will be used to instantiate the work models, the specific learning objects are ready to be designed. In addition to providing meaningful differentiation between learning object types, the learning object taxonomy presented above can be read alternatively as a set of guidelines for the design of learning objects. Overarching considerations of development and delivery cost and time must be taken into account.

Designing Single-type Learning Objects. Single-type learning objects should be designed in such as way as to function in the greatest number of contexts possible. They should consist of an individual element of a single media type. That is, two photographs presented together in a single file would constitute a Combined-intact learning object, even though there is only one file and one media type. An example of a Single-type learning object would be a scanned image of the Mona Lisa. Single-type learning objects that are text-based should still adhere to the constraint of serving the Exhibit function. Quotations are usable as Single-type objects.

Designing Combined-intact Learning Objects. Combined-intact learning objects should be designed to present a single, stand-alone, whole piece of information. While any combination of media may be considered in the design of Combined-intact objects, Combined-intact objects should be restricted to the combination of two to four elements, as statically combining a larger number begins to work against the principle of reuse. Limited logic can be built into Combined-intact objects, and many static combination environments (QuickTime, Director, ToolBook, etc.) have scripting languages which can be used to create simple item scoring. While the Combined-intact object cannot be used in as many arbitrary contexts as the Single-type, care should still be taken in design to not unnecessarily limit the number of possible contexts of use. Examples of a Combined-intact learning object include a map of Sherwood Forest (the static combination of an image with text labels) and a digital movie (the unrecoverable combination of video and audio).

Designing Combined-modifiable Learning Objects. Combined-modifiable objects seem deceptively simple to design, as they are “just” the combination of other learning objects. However, the purpose of Combined-modifiable is to be “instructional,” and much of the hardest work of design goes into the design of Combined-modifiable objects. This is because Single-type and Combined-intact objects can rarely simply be sequenced in an instructional manner when they have been designed in such a way as to promote inclusion in the greatest number of contexts possible. Meaningful transitions from one object to another must be effected (textual and otherwise). Introductions and summaries must be crafted in ways that support encoding and elaboration. Motivational strategies must be designed and integrated. Each of these types of information is extremely non-reusable. For example, an introductory section that includes cognitive scaffolding for the skills within a work model *is only usable with that specific combination of skills and associated learning objects*.

Having said that Combined-modifiable objects contain information that would be less reusable individually (such as a transition language between two objects), Combined-modifiable objects can still be designed in such a way as to make them reusable as a whole. For example, while the potential for reuse of transitional material between two objects is low, the potential for reuse of a Combined-modifiable object that teaches an entire work model is relatively high. Although a the transition between a history of the Mona Lisa and an exposition of the artistic qualities of the Mona Lisa is far less reusable than the image alone, the entire Combined-modifiable object containing the image, history, and exposition could be used in many ways.

Designing Generative-presentation Learning Objects. Generative-presentation learning objects contain additional design considerations previously unnecessary in the design of learning objects. Generative-presentation learning objects should be designed accordingly to guidelines similar to Combined-modifiable with the additional considerations of logic and data files. Abstract logic must be created for the layout of the objects (rules regarding the on-screen positioning of constituent objects). The logic must be created with the data file already in mind, as the parameters and arguments of the data file must be parsable and usable by the logic for the Generative-presentation learning object to create the specific presentations.

Designing Generative-instructional Learning Objects. Learning environments based on Merrill's (1999) Instructional Transaction Theory are the best existing examples of the Generative-instructional learning object type. (An instructional transaction is all of the interactions that must occur for a student to achieve a certain type of learning goal, such as remember a list, follow a procedure, classify, etc.) Because of Instructional Transaction Theory's complexity, and because writing new design guidelines for the Generative-instructional type would, effectively, be rewriting existing work of Merrill's, Instructional Transaction Theory will be briefly overviewed here.

Instructional Transaction Theory (ITT) has four basic goals: the creation of effective instruction, increasing the efficiency of instructional development, producing instructional simulations and microworlds, and providing adaptive instruction (Merrill, 1999). Merrill claims that any instructional strategy, whether algorithmic or experiential, can be described in terms of ITT methods. Merrill (1999) describes the components of a learning environment based on ITT as follows:

> A learning environment is described in terms of (a) the instructional goal it is designed to promote; (b) the knowledge structure required by the learning environment; (c) the general simulation engine which operates on this knowledge structure to represent activities and processes that occur in the world; and (d) the learning activity of exploration by which the student interacts with the learning environment.

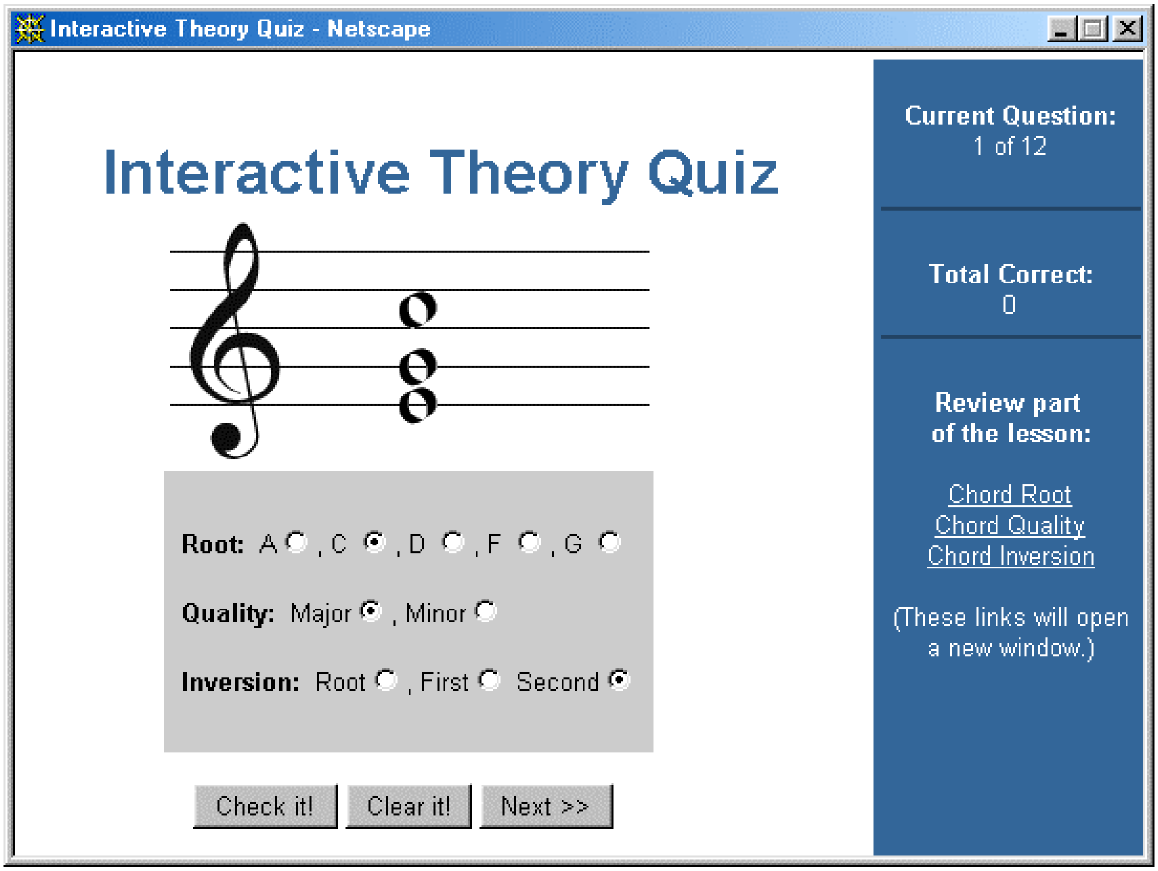
This type of learning environment is similar to that described for Generative-presentation, except that the layout and evaluation logic always take the form of a highly generalized simulation engine with embedded instructional strategies. Merrill, Li, and Jones (1992) identified thirteen classes of instructional strategies or transactions, including identify, execute, interpret, judge, classify, and generalize. When a knowledge structure (comprised of several "knowledge objects") is constructed in a manner consistent with the simulation engine architecture, the result is an instructional simulation capable of generating instruction, problems, and evaluating answers.

Early versions of software built around Merrill's theory (i.e., the IDXelerator) used the knowledge object architecture, but the instructional simulations were built in such a manner as to make their components unrecoverable, and therefore non-reusable. While such a simulation has a high potential for intra-contextual reusability, the fact that the large number of necessary component parts are non-reusable (unrecoverable) makes the development investment much more costly than need be. When a software implementation of ITT evolves which can access, utilize, and wire together (in the JAVA Beans sense of the term) knowledge objects represented in a standard manner (e.g., in a network accessible XML document and various representational forms), Generative-instructional learning objects may be more tightly integrated into LODAS.

The following example in which Cheryl plans to use and designs learning objects is by no means ideal. Cheryl is working without a development or delivery system designed to support learning objects, and with a single graduate student programmer. Ideally, LODAS would be deployed in connection with a system that archives objects, manages their metadata, and delivers them in user-specified combinations. However, this technology is still “coming,” and so Cheryl and her graduate student find themselves doing things by hand in the “real world.”

Example. This is new ground for Cheryl who begins planning to use learning objects with some excitement. There is no learning object database (that she knows of) to choose from, and so she is left to design everything from scratch. First, she considers the notion of creating everything as a Single-type object. This seems extreme in the case of text, and she decides to create Single-type objects for illustrations only. Next, she considers how to archive her objects. She has neither the time, money, nor understanding of metadata standards to create formal metadata records for each of these objects, and her university has yet to deploy a system that would support her learning object effort, so after some consultation with her graduate student programmer she decides to create a spreadsheet to store one important piece of information about each Combined-intact object: the full path to each constituent object on her network drive. This will help her find the Single-type objects again next semester when she comes back to redesign. With this decision made, Cheryl goes on to plan what type of learning objects she will use with the examples and practice she has just outlined.

For the “Identify the root, quality, and inversion of chords” work model, Cheryl begins by planning to use combined-intact learning objects for the JIT instruction and job aids, and a Combined modifiable learning object, which will use a Generative-presentation learning object to present the problems, for the set of specific problems she has identified. She sets about specifying the chord analysis algorithm to her graduate student, who codes it. He additionally codes a random value generator that picks a root, quality, and inversion from provided parameters, uses the chord analysis engine to build the chord, and then uses Single-type staff and note representations to visually present the randomly generated chord to the student for analysis. His initial prototype looks like Figure 6. He happily announces that creation of new parameter files (simple text files from which the learning object gets the parameters for the generation of random chords) is all that will be necessary to create the increasingly complex case types for this work model. Feedback, etc., will be delivered through pop-ups like that shown in Figure 7. Cheryl is also very pleased.



Interactive Theory Quiz

Root: A ○ , C ● , D ○ , F ○ , G ○

Quality: Major ● , Minor ○

Inversion: Root ○ , First ○ Second ●

Current Question:
1 of 12

Total Correct:
0

Review part
of the lesson:

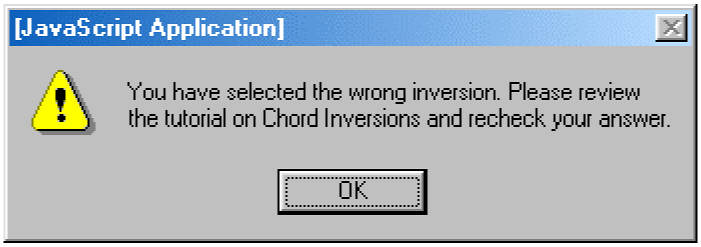
Chord Root
Chord Quality
Chord Inversion

(These links will open
a new window.)

Figure 6. Combined-modifiable learning object. The case type template for the "Identify the root, quality, and inversion of chords" work model.

Cheryl steps back to take stock of her planning. She has created the specific problem elements (the staff and notes) as Single-type learning objects as suggested.

These are encapsulated in a Generative-presentation object. Finally, she has created a combined-modifiable template in which to combine and deliver the objects, as suggested. Everything seems to be going well.



You have selected the wrong inversion. Please review the tutorial on Chord Inversions and recheck your answer.

Figure 7. Feedback mechanism. Sample feedback that would be provided for the answer given in Figure 5.

Cheryl realizes the benefits of reusable components when she starts to create the worked example Generative-presentation object – it looks just like the problem object with two changes. First, correct information as generated by the random value generator is displayed in place of the current answer entry form. Second, the right frame contains the prerequisite information necessary to complete the example. The Generative-presentation object designed for practice is usable with a minor modification. (The graduate student is even happier than Cheryl at this point.)

On reviewing the JIT instruction, Cheryl finds it comprised of procedural job aids, definitions, and explanations. These are probably text-based, which would make them combined-intact objects. If she chooses to add graphic examples, she can combine one of the Single-type objects created for the examples or practice with her instruction for a combined-modifiable object.

Design Learning Object Sequencing

Overview. This group of methods functions together to provide instructional sequencing specifications for the learning objects designed previously.

At this stage of the design process, the content domain has been both analyzed and synthesized, the instructional design has been completed, and learning objects have been designed based on the instructional design. The final step in the design process is the design of the learning object sequencing. Sequencing of learning objects must occur at three levels: sequencing of work models, sequencing of case types, and sequencing of specific problems. Guidelines for the three types of sequencing are presented in this section.

Work model sequencing (Bunderson, Newby, and Wiley, 2000; van Merriënboer, 1997). In the content analysis and synthesis process described above, individual work models were developed and placed in order of increasing difficulty along unidimensional scales of expertise (as illustrated in Figure 4). Additionally, each scale was divided into sections roughly corresponding to levels of expertise during this synthesis process (see Figure 4). The learner should complete each of the work models at a given level before attempting work models at the next level. Thus, the sequencing design is partially completed, and the remaining task of sequencing work models occurs across scales within each of the levels. That is, the sequence of delivery of work models within the Novice level must be considered separately from the sequencing of those at the Intermediate level, and so on.

Begin by identifying any dependencies between work models at each level. Learning objects should be sequenced according to any such dependencies discovered.

The remaining learning objects, i.e., those without such dependencies between work models, should be sequenced in a random fashion (Paas and van Merriënboer, 1994a, 1994b), as "increasing the variability or contextual interference of the problem sequence will substantially enhance inductive processing and transfer" (van Merriënboer, 1997; p. 190). When the learner has completed each of the work models at a given level, all learning objects designed to teach integrated work models for that level should be delivered in a random sequence.

Additionally, constituent skills within each work model must be sequenced. This sequencing should follow the same guidelines laid out for work model sequencing, namely, a simple to complex ordering that respects dependencies between the skills.

<u>Case type sequencing</u> (Reigeluth, 1999; van Merriënboer, 1997). While case types themselves are never directly presented to the learner, problems and worked-out examples based on the case types are. In other words, every problem and worked-out example is an instance of a case type. Case type sequencing deals with the sequence in which these groups of problems will be presented to the learner.

Case types within a work model should be sequenced in a simple to complex ordering. Because each case type should be designed to represent the entire work model to be learned (with varying levels of simplifying conditions applied), the sequencing of case types almost exactly resembles the Simplifying Conditions Method of Elaboration Theory (Reigeluth, 1999) explained previously.

<u>Specific problem sequencing</u> (van Merriënboer, 1997). Instantiated, a case type is a group of several learning objects that present the specific problems to be practiced at one level of expertise within a work model. While the previous activity sequenced these

groups of problems, the individual learning objects within each case type must now be sequenced.

The learning objects that present the specific problems within a case type should be ordered randomly in order to create contextual interference, as explained above. Because the specific problems are instances of a case type, which represents a single level of expertise for the whole work model, each of the learning objects will be instructionally equivalent. The difficulty in promoting transfer comes from the fact that the individual problems may be built around a few building blocks and ordered accordingly. For example, for the case type “Chord identification,” three inversions and two qualities may be used to create six problems. The designer’s instinct may direct a problem sequence like the following: root inversion / major, first inversion / major, second inversion / major, root inversion / minor, first inversion / minor, second inversion / minor. However, taking the opposite approach and randomizing the delivery of these problems will promote greater inductive processing and transfer than the “ordered” sequencing.

Example. Cheryl now finds herself confronted by a large number of learning objects (representing work models and their components) that need to be sequenced. She knows to deliver the work models comprising the first integrated work model(s) first, followed by the integrated work model(s), and so forth. So she decides on the following macro-sequence: Chords, Cadences, Integrated 1 (Cadence Classification), Non-chord tones, Motives and Phrases, Integrated 2 (Phrase Reduction), Harmonic Function, Periods, Integrated 3 (Analyze a Piece of Music).

One level down, Cheryl also needs to sequence the constituent skills within each work model. The Chords work model includes the three constituent skills Intervals, Stacking in Thirds, and Chord Identification. This order already represents the dependencies or prerequisite relationships between the skills, and Cheryl leaves them as is. She then repeats this process for the constituent skills contained in the remaining work models.

Cheryl now needs to sequence case types within each constituent skill. She knows to order each of the case types within each constituent skill in a simple to complex manner, and begins to do so. In the case of Intervals, the first constituent skill of the Chords work model, case types using the treble clef are ordered before those using the bass clef. For Stacking in Thirds the same will hold true, case types in root position will precede those other inversions, and case types with three notes will be presented before those with four. Finally, for Chord Identification, treble clef case types will again precede bass clef case types, major case types will precede minor ones, and case types in root position will again precede those other inversions.

Cheryl breathes a sigh of relief… she's almost home. The final step is to sequence the specific practice items within each case type. Imagine her joy when she learns that this sequencing is random!

Cheryl has completely implemented the LODAS theory. While her graduate student will have to do some clever programming while the technology catches up with Cheryl's instructional design methodology, she has applied the theory unaltered, and will be ready for the new technology when it becomes available.

Loop Back for Quality Improvement

Once instruction and learning objects have been designed and developed, three important steps in the theory implementation process remain: implementation, evaluation and revision. While the final chapter of this study deals with evaluation and revision of the LODAS theory itself, the final step of the LODAS methodology relates to the implementation, evaluation and revision of the instruction and technology created through the instantiation of the theory.

The quality improvement loop should be seen as an ongoing activity. Quality improvement is part of every aspect of a design theory. The metrics for quality improvement are specified at the beginning of work, and these guide both the design and the ongoing evaluation. Formative evaluations (such as usability tests) can be conducted using individual and small groups during the development process.

A micro-design experiment approach (Brown, 1992; Bunderson, 2000) to instructional quality improvement is one possibility. In a typical design experiment, a single research cycle might run an entire semester; however, micro-design experiments run in shorter cycles and several can be completed within the context of a single design experiment. In a micro-design experiment framework for instructional quality improvement, an instructor would begin by carefully observing the function of their instruction and noting weaknesses (e.g., a new instructional strategy which does not seem to communicate with students). Design changes or interventions intended to strengthen these weaknesses are introduced in the next micro-design experiment cycle (instead of at the end of the semester) so that students can benefit from this formative evaluation as soon as feasible. This will likely include the just-in-time redesign of instruction.

Example. Cheryl decides that she wants to begin quality improvement work right away, and offers lunch to several of her best students from the previous semester if they will spend a day with her formatively evaluating the new instruction prior to the start of classes. Cheryl asks the students to interact with her prototype learning objects and solicits their feedback. Cheryl views this pre-class testing as cycle 0 of several micro-design experiments that she will run during the semester, and incorporates the initial user feedback prior to the first day of class.

Summary

Learning Object Design and Sequencing Theory (LODAS) is a combination of methods synthesized from several existing theories and new work done on a taxonomy of learning objects. Synthesized, these design strategies comprise a new instructional design theory that begins with a content domain and ends with instructional strategies instantiated in learning objects, which in turn are sequenced according to research on transfer of skills from the learning environment to the environment of practice.

Now that LODAS has been described in detail sufficient for implementation, the final chapter will outline future research directions, including a method for testing and improving LODAS over implementation cycles of a design experiment.

CHAPTER FIVE

ACCOMPLISHMENTS AND FUTURE DIRECTIONS

This study represents an effort to place a new instructional technology called "learning objects" within the context of instructional design theory. While Learning Object Design and Sequencing Theory (LODAS) is the only known instructional design theory to provide explicit support for the instructional design and instructional use of IEEE 1484 compatible learning objects, it is by no means perfect. Speaking of the instructional design theories contained in Reigeluth (1999) such as Schank's Learning by Doing and Merrill's Instructional Transaction Theory, Reigeluth and Frick (1999) state:

> But none of the theories described in this book has yet been developed to a state of perfection.... it should be patent that the development and testing of design theories is not a one-trial endeavor. It is a matter of successive approximation. Such theories continue to be improved and refined over many iterations" (p. 633-635).

This study has presented the first iteration in the theory development process for LODAS. It has attempted to accomplish the stated goals of the theory's creation; however, as Reigeluth and Frick are so quick to point out, theory development is an iterative, on-going process, and LODAS needs to be "improved and refined over many iterations." This chapter will first review the original goals of the study and the manner in which LODAS attempts to accomplish them. Finally, a theory-testing or theory-improvement plan outlining future research will be presented.

Accomplishing the Goals of the Study

Chapter 4 states two groups of goals for LODAS, purposes which the theory should serve and desired characteristics of the theory itself.

Purposes of the theory. The first four goals, which relate to the functionality of LODAS, are as follows:

1. Catalyze dialog around the use of learning objects in an instructional design context,
2. provide explicit support for the design of learning objects,
3. provide explicit support for the sequencing of learning objects, and
4. provide learning object support in a reusable manner, and
5. provide forward compatibility with expertise-based domain and learner modeling research.

The entirety of Chapter 4 addresses the first goal. To date the dialog in the field surrounding learning objects has been almost exclusively technical in nature. Whether the instructional design theory presented in Chapter 4 is widely adopted or not, if it catalyzes a shift of dialog in the field away from technology standards and toward the instructional use of learning objects, it has accomplished this first goal. The funding of a grant by the National Science Foundation to continue this work, as well as the Association for Educational Communications and Technology's (AECT) acceptance of a proposal to publish an edited book based on many of the ideas presented in LODAS, will provide the first opportunity for this catalyzing to occur.

The learning object taxonomy and group of methods labeled **Select and / or Design Learning Objects** presented in Chapter 4 address the second goal. These methods function together to specify what types of learning objects should be designed to achieve certain instructional ends and how different types of learning objects should be designed.

Likewise, the group of methods labeled **Design Learning Object Sequencing** address the third goal. These methods function together to prescribe a sequence for the learning objects selected or designed previously.

Finally, the group of methods labeled **Analyze and Synthesize Content** address the last goal. By designing a method of describing the domain in terms of expertise instead of "knowables" the door is left open to continuing research in both domain mapping and expertise-based learning modeling, in terms of their position on the domain map.

Characteristics of the Theory. Chapter 4 also set a goal for the theory to bear certain characteristics. While verification of the existence of each of the characteristics identified in Chapter 4 is an ongoing endeavor, the current status of this verification is provided below.

- Significance – The theory is the first to provide explicit support for the design, scope, and sequencing of IEEE 1484.12 compliant learning objects. The theory presents a unique method for getting from an undifferentiated content domain to a multi-dimensional domain map with qualitatively ordered constructs of expertise. Perhaps most importantly, it provides a general taxonomy of learning objects that can be linked into existing instructional design theories and demonstrates one example of such a linking.

- Communicability – The theory has proven communicable insomuch as it has been captured in writing and illustration. The imminent publication of the theory will present its first independent test of it communicability.
- Parsimony – The author has completely implemented the theory conceptually, as presented in Chapter 4, and worked many of the unnecessary complexities out of the theory. Experts in the field have read several drafts of LODAS and communicated ways in which both the theory and its exposition could be simplified. These recommendations have been largely implemented. While the current version of LODAS may still not be describable as "parsimonious," it is much more so than the earlier versions. As the theory is implemented and more clearly understood, this understanding will be reflected as greater parsimony of new versions of the theory.
- Realizability – Because very few software tools currently exist that support the learning object paradigm generally, let alone the specific version presented in LODAS, the author is now engaged in building software tools to support the technology portion of implementation (this work is being funded partially by NSF). As reported earlier, the theory has been completely implemented conceptually, and the theory was found to be implementable technologically as well as conceptually.
- Sustainability – As the following section will present, a rigorous research and theory-testing or theory-improvement agenda has been built around the theory. This provides a formal method for the testing and improvement of the theory, sustaining its development.

While LODAS meets the goals originally set in its current state, the degree to which it meets these goals can be improved through an iterative process of formative research and theory-testing. The last section of this chapter presents a future research agenda, including theory-testing and theory-improvement studies that can be used to improve the theory.

Testing and Improving the Theory

Accepting Reigeluth & Frick's (1999) statement that a design theory never achieves perfection, the motives for gathering evidence regarding the performance of a design theory necessarily move from the traditional educational measurement view of measuring-to-sort-into-categories to a view more compatible with the new paradigm of instructional design theory identified by Reigeluth (1999), measuring-to-diagnose-and-improve. While this paradigm is relatively new to instruction, it is not new to other fields (c.f., Deming, 1995 and other sources on Quality Improvement). Accordingly, this section will outline a theory-testing and improving framework by which diagnostic evidence can be gathered for the purpose of improving the theory over multiple iterations.

<u>Improving the theory's methods</u>. Testing a design theory at the global level with a preferability study (Reigeluth & Frick, 1999) can be seen as a type of product evaluation (Bunderson, Martinez, & Wiley, 2000). This type of study describes the function of the theory holistically but reveals little about how the individual design prescriptions or methods are actually functioning. If the instructional design theory is not functioning as desired, it may be impossible to isolate the offending method(s). On the other hand, testing an entire instructional design theory at the level of the individual design prescriptions or methods (looking for improvement opportunities) can be difficult, time

consuming, and *unnecessary*, if other less time-consuming, but equally powerful, methods of testing are available to the theory developer.

A moderate approach will be proposed here to initially test the methods of LODAS: testing groups of methods that function together. If the overall function (such as **Design Learning Object Sequencing**) is being performed in a manner satisficing to the developer, questions regarding the basic quality of the theory can be put behind and the developer can move on to focus on improving the theory. The "satisificing" level will differ depending on the instructional environment, e.g., the satisficing degree of instructional effectiveness for nuclear reactor training will likely be much higher than the degree for computer software usability testing. However, if a group of methods is not performing its function in a satisficing manner, the individual methods within that group can be tested individually.

Design experiment framework. In a design experiment framework (Collins, 1990; Brown, 1992; Bunderson, 2000), data from previous research cycles can serve as control data for later cycles. Bunderson (2000) has argued that a construct-valid, invariant growth scale is actually superior to randomization as it provides the experimental option of matching subjects (instead of depending on randomization). Domain theory (Bunderson, Newby, and Wiley, 2000) provides the invariant measurement framework in which meaningful comparisons can be made across cycles of different people and different measurement tasks or items. This measurement framework (which is already compatible with LODAS because it was one of the theories from which LODAS was synthesized) and the design experiment notion of comparison across cycles both fit within Reigeluth and Frick's (1999) view of the iterative nature of instructional design theory improvement and provide the foundation for the theory testing plan, and principles generally accepted in the field of Quality Improvement (Deming, 1995).

Unified validity framework. A unified validity methodology will provide the framework for testing LODAS, following Bunderson, Martinez, and Wiley's (2000) extension of Messick's unified validity framework. Messick (1998) describes the process of "constructing construct validity," following Cronbach's (1988) assertion that validity is never established *per se*. Rather, evidence is gathered and combined with rationale to create an ongoing "validity argument" that gains strength over time. A validity argument is a combination of evidence and rationale, in many ways similar to the argument a trial lawyer might present in court. In the case of testing and improving an instructional design theory, evidence regarding the function of the theory and its methods is gathered and combined with explanatory rationale. However, the theory developer has a significant advantage over the trial lawyer: when evidence is found against the instructional design theory, the theory developer can make principled changes to (i.e., improve) the theory and gather new evidence. This cycle of gathering evidence and building rationale around the evidence is the process of creating a validity argument.

This view also fits with Reigeluth and Frick's (1999) description of the ongoing nature of theory development and improvement. The Bunderson, Martinez, and Wiley (2000) extension of Messick's framework extends the number of types of evidence gathered for the validity argument to seven. These seven types of evidence from Bunderson, Martinez, and Wiley (2000) are presented below. (If this framework seems somewhat generic, it is purposefully so. Bunderson, Martinez, and Wiley do not propose specific tests to be carried out in the testing of design theories, but they suggest a unified validity framework in which the testing can occur. The general framework is presented here for the reader's reference; the specific theory-testing plan for LODAS is presented in the following section **The Theory-Testing Plan**.)

1. Content and Substantive Process -- Establish the representativeness, appropriateness, and completeness of the content. Simultaneously, establish the first level of understanding of the substantive processes of growth in the domain. These steps are accomplished through a test-blueprint document and rationale. For this theory, the test-blueprint must include as substantive processes the various constructs of growth, as well as the appropriate content.

2. Structural Aspects – Establish the dimensionality and boundaries of the domain through factor analytic and other studies, and conduct scoring studies to assure that the score structure accurately reflects the domain structure. Place growth constructs or work models on the scales (dimensions) in an order reflecting the expertise necessary to successfully complete the tasks.

3. Generalizability Aspects – Determine whether domain content is represented sufficiently within instruction and measurement instruments to facilitate generalization across tasks within the domain. Determine whether sample and occasion representativeness facilitate generalization to other groups of people and different measurement occasions.

4. Convergent and Discriminant Studies **–** Design or find alternative measures of the same construct and compare measurement outcomes across instruments, people, and occasions. Measures of the same construct should converge to provide triangulated evidence for the construct. Identify plausible rival hypotheses and design or find alternative measures of these alternative constructs. Measures of rival hypothetical constructs should be unrelated to the construct of interest, allowing the researcher to discriminate between constructs.

5. Consequential Aspects **–** Develop evidence and rationales for judging the short- and long-term consequences, both intended and unintended, of score interpretation and use. Ensuring that negative consequences for groups or individuals not be the result of invalidity on the part of the instruction or test is traditionally the major focus of this activity. However, in the context of instructional design theory testing, determining whether or not intended positive consequences occur (such as students reaching desired levels of expertise) becomes the most significant consideration.

6. Efficient Utility – Efficient utility refers to the relative costs and benefits of instruction and testing, and is strongly linked to the manner in which material learned or test scores will be used. Generally, the anticipated benefits of designing and delivering the instruction, designing and administering the instrument, and gathering and analyzing the data should outweigh the costs of doing so. Efficient utility encompasses and extends Reigeluth’s (1999) metric of *efficiency*.

7. Value Implications – Value implications relate to students’ perceived value of the instruction and practice resulting from application of the theory. Considerations of relevance, face validity, and simulation fidelity are part of value implications studies. Value implications encompasses and extends Reigeluth's (1999) metric of *appeal*.

While these seven areas are meant to be comprehensive, not every type of validity evidence will always be brought to bear on every research question. However, as several research questions are asked and answered, and as different types of evidence regarding the performance of the theory are gathered, a strong validity argument can be constructed from the diverse resulting evidence.

<u>Research questions</u>. LODAS contains four major blocks of methods that can be tested: Analyze and Synthesize Content, Design Practice and Information Presentation, Select and / or Design Learning Objects, and Design Learning Object Sequencing. While many questions could be asked regarding the functioning of each of these blocks of methods, the following theory testing plan will focus on the following four questions (each related to one block of methods).

1. Do the "Analyze and Synthesize Content" methods function together to produce a construct valid map of expertise within the domain of interest?
2. Do the Four Component Instructional Design Model instructional strategies in the "Design Practice and Information Presentation" methods continue to facilitate learners' integration of constituent skills in the context of the new analysis and synthesis procedure and learning object implementation?
3. Do the learning object types recommended in the "Select and / or Design Learning Objects" methods facilitate attainment of the identified instructional purposes?

4. Do the sequencing strategies in the “Design Learning Object Sequencing” methods facilitate instructional appeal on the part of the students?

The selection of these four questions does not mean that they are the only worthwhile questions to be asked regarding LODAS. However, these questions are important ones to be asked regarding each of the blocks of methods, and should serve to reveal weaknesses in the function of the blocks.

The Theory Testing Plan

Having established the components of the theory testing methodology (i.e., a design experiment framework, a manner of blocking methods into testable groups, research questions anchored to the method blocks, and a method of building a unified validity argument), the stage is set to lay out the specifics of the theory-testing plan. For the reader’s reference, numbers are included parenthetically to reference the types of validity evidence used in answering each question.

The theory-testing plan laid out hereafter focuses on the initial investigation of the theory, in order to assure that the theory is worth the time that must be invested in order to improve it over several research cycles. While this data may also be used to discover ways in which the theory can be improved, the primary purpose of these specific questions is to provide a method of falsification of the theory’s methods. This theory-testing study employs a “balanced attack” of mixed methods, using both quantitative and qualitative methodologies to answer two questions.

Question one. The first research cycle should focus on the first block of methods (Analyze and Synthesize Content), not because it is first, but because problems with the domain map produced as a result of this block of methods will confound the accurate answering of other research questions. Of interest is the first question:

> Do the “Analyze and Synthesize Content” methods function together to produce a construct valid map of expertise within the domain of interest?

This question is best answered with studies that examine (1) Content and Substance Process and (2) Structural Aspects of the domain. The “substantive growth processes” in the domain are constructs in the domain map to be validated. These may be cognitive, linguistic, motor, or other processes involved in the work models (which require activities whose successful performance indicates a certain level of expertise in the domain). The work models in turn are ordered along scales according to their difficulty. The “structural aspects” of the domain refers to the dimensionality of the domain. The dimensions or scales together with the ordered work models comprise the “content” of the domain.

After identifying the dimensions or scales of expertise within the domain, creating work models anchored to the scales in a specific order (see Figures 3 and 4 of Chapter 4), and designing practice items whose specifications are the work models (or in other words, after completing all the methods in this block), three parallel versions of an instrument designed to test expertise in the domain can be created. These can be administered in a pre-test, midterm, and post-test fashion, and will contain items which should be parallel due to their being specified by the same work models. Data from these tests can be analyzed in two ways to provide important information regarding the content, substantive growth processes, and structure of the domain.

A factor analysis of the results from the tests will provide data (specifically, a number of factors) that can be used to confirm or disconfirm hypotheses regarding the number of dimensions of expertise within the domain and to which dimension individual work models belong. Comparison of factor analyses across tests can provide stability of interpretation. Membership information for each item (information regarding the factors on which the individual items load significantly) can provide information to confirm or disconfirm hypotheses regarding the anchoring of a specific work model on a specific dimension of expertise. This information can also be used to induce the identity of unpredicted factors as the group of items that load significantly on the factor is examined. This information *cannot* be used, however, to test hypotheses regarding the ordering of items (according to difficulty) along the scale. (When factor analytic techniques cannot be used, tests of dimensionality can be performed using the Rasch model.)

In order to answer questions regarding difficulty orderings of work models along dimensions of expertise, item response theory methods can be used to calculate difficulty estimates for each of the items. Again, multiple items (items on multiple tests) designed from the same specification (work model) will provide stability of interpretation. The specific difficulty estimations are not of interest so much as the qualitative ordering of difficulty estimates of items whose specifications are separate work models. This information can be used to confirm or disconfirm hypotheses regarding work model orderings. Additionally, this information can be compared for items developed from a single work model to examine the degree to which parallel item forms are truly parallel.

This general approach to testing content and substantive growth processes and structural aspects has been demonstrated successfully by Strong-Krause (2000) and Bunderson, Martinez, and Wiley (2000).

Any problems with the domain map that can be corrected at this point, e.g., improper work model orderings, should be corrected before proceeding to the next step of theory-testing.

Question Two. The data gathered for the previous analysis can be reanalyzed to answer the second question:

> Do the Four Component Instructional Design Model instructional strategies in the "Design Practice and Information Presentation" methods continue to facilitate learners' integration of constituent skills in the context of the new analysis and synthesis procedure and learning object implementation?

This question can be answered with a (5) Consequential Aspects study, in which the effectiveness of the instruction (the intended positive consequence of the instruction) is measured.

Confirming or disconfirming evidence for this question can be gathered by correlating learner scores on individual work models with learner scores on the integrated work models. We would expect learners who struggle with the individual work models to struggle with the integrated work models, as they require the synthesis and application of these constituent skills. Likewise, we would expect learners who excel with the individual work models to excel with the integrated work models. Positive correlation of work model scores to integrated work model scores would indicate confirming evidence for the integrative function of the methods.

More simply, the data may be divided into four groups that can be inspected visually to reveal weaknesses in the instruction. Such a division is represented in Table 2.

Table 2. Sample Consequential Aspects Evidence.

	Failed Integrated Work Model	Passed Integrated Work Model
Failed Constituent Work Models	5 A	0 B
Passed Constituent Work Models	5 C	39 D

Note. The number in the center of each cell represents a number of students. The capital letters in the bottom right of these cells are for reference in the discussion.

If the amount of students in cell A (i.e., the number of students who fail both the constituent work models and the integrated work models) is high, the instruction is not functioning at all, let alone functioning to facilitate integration of constituent skills. If any students appear in cell B (meaning that they were able to successfully complete the integrated work models without mastering the constituent skills), this most likely indicates a problem in the description of the domain, or more specifically, the domain map. Cell C is the key to answering research question two. If the number of people who can complete the constituent skills but cannot complete the integrated skills is large, this may indicate a problem. Ideally, every student in the class should be in cell D.

Question three. Additional data will be necessary to adequately answer the third question:

> Do the learning object types recommended in the “Select and / or Design Learning Objects” methods facilitate the identified instructional purposes?

Quantitative data have already been analyzed for Question Two in order to address issues of the (5) Consequential Aspects of the instruction. This information described student ability to perform on the individual and integrated work models. Student performance is the strongest quantitative measure of whether or not learning objects are getting the instructional job done. Therefore, a qualitative (7) Value Implications study will be used to gain additional insight into the reasons why the instruction has been created in the manner it has.

The quality of the learning objects' instructional function has already been quantitatively measured in order to answer Question Two. Triangulating evidence can be established through interviews with the instructional designers who designed the instruction. Questions regarding the affective experience of designing learning objects within the LODAS context will reveal two important types of information. First, assuming that instructional designers "know best" and will do what seems correct regardless of what theories dictate (Rowland, 1992), a general sense of easiness and agreement with the design guidelines and their application will provide evidence that the prescriptions for learning object design are working well. A general sense of frustration or fruitlessness from the designers will provide evidence of difficulty in applying the learning object design guidelines.

Second, and perhaps more importantly, gaining an understanding of the shortcuts taken and situation-specific adjustments made during the application of the theory will provide precious insights into the real-world application of LODAS and valuable formative data for use in its revision. Additionally, if the instructional designers have deviated from the methods and can articulate the manner in which they did, these data may be compared later with data from a pure implementation (as may Question Two data) for (4) Convergent and Discriminant studies that begin to show the differential functioning of the individual methods and the block of methods as a whole.

Finally, if Question Two and Question Three data are both positive (qualitatively speaking), this provides strong mixed-method triangulation for the success of LODAS. This is the most powerful kind of evidence available to the constructer of a validity argument.

Question four. Finally, additional qualitative data will be needed to address the fourth question:

> Do the sequencing strategies in the "Design Learning Object Sequencing" methods facilitate instructional appeal for students?

"Transfer is not what I would consider the most important effect of sequencing strategies. I would put appeal first, ease of learning second (one dimension of efficiency), long-term retention third, and transfer fourth" (Reigeluth, personal communication, June 14, 2000). Questions of appeal are also best answered by a (7) Value Implications study. Reigeluth and Frick's (1999) three metrics for judging instructional design theories are "effectiveness, efficiency, and appeal" (p.635). Research questions two and three in this chapter have addressed questions of LODAS' *effectiveness*. Questions of *efficiency* are best left until a digital library of learning objects exists, since early attempts to apply LODAS (without access to a digital library) will require the creation of all learning objects from scratch. Research question four addresses Reigeluth and Frick's criterion *appeal*.

A Likert-style attitudinal instrument (a Kirkpatrick level 1 evaluation) can be used to answer this question. Items on the instrument should focus on students' overall affective experience with the instruction and their affective experience with regard to the order in which the instructional materials were presented. While selected response items may be sufficient to gather the necessary data, open-ended questions may also provide feedback useful for the improvement of the theory.

Instructors may also create voluntary approach and avoidance opportunities for students in order to determine the degree to which students are voluntarily engaging the instruction. This information could be stored by the instructional system itself (Martinez, 1999) or kept by the instructor in a log of incidental comments, experiences, and anecdotes that occur over the course of the instruction.

Questions regarding appeal may need to be asked and answered first if students refuse to engage the instruction because it lacks face validity, simulation fidelity, or obvious relevance. These problems could potentially prevent the collection of any data, necessitating the investigation of appeal-related issues first in the research cycle.

When things go wrong. A methodology for testing blocks of design methods has been presented and exemplified in a theory-testing plan for answering four significant questions regarding the new instructional design theory. However, occasions will certainly occur when answers to questions do not meet expectations and methods must be tested individually. While testing individual methods with (4) Convergent and Discriminant studies are possible (as described above), two important prerequisites to such testing are difficult: identifying the malfunctioning method and selecting a candidate for replacement. This brief section presents a small set of guidelines to follow when completing these tasks.

If the theory developer had access to no information other than the fact that “the block of methods was malfunctioning,” each method would need to be experimentally isolated (to the degree possible) and tested, or tested within the micro-design experiment framework. However, there will frequently be clues available that can help narrow the search for errant methods. These clues may be gathered through qualitative studies involving interviews of designers, instructors, and learners. Generally, one member of this group will be able to identify something that “didn’t quite feel right” in the design, delivery, or interaction with the instruction. This method or sub-group of methods should be the starting point of further studies.

More importantly, however, designers, instructors, or learners may be able to identify ways in which practice deviated from the theory. These are implementation fidelity issues. The theory developer must then ascertain whether or not the deviation was the cause of the breakdown. This situation can alert the theory developer to critical conditions (of the goals, conditions, and methods of instructional design theory) that may have been overlooked in the development of the theory.

Finally, in order to identify alternative methods for use in (4) Convergent and Discriminant studies, the theory developer can rely on past experience, reviews of pertinent literature, or even sensible “intuitions.” These methods can then be used as alternatives in the context of (4) Convergent and Discriminant studies as described above with either traditional experimental designs (when possible) or design experiments.

Conclusion

This chapter has presented a methodology for testing and improving LODAS. The methodology has been exemplified in the first four questions that might be asked regarding the theory’s performance and potential for improvement, and additional questions can be asked within the same general framework. The design experiment framework provides an additional experimental design method that allows previously “unaskable” questions to be asked.

As diverse evidence is gathered through the application of different types of unified validity studies to different research questions, a rich body of qualitative and quantitative confirmatory data can be accumulated. These data can be combined with explanatory rationale to create a validity argument regarding the quality of LODAS.

As noted above, the theory will be published in the Fall of 2000, increasing the possibility of its widespread application and opportunity for improvement. Additionally, the author has been offered the opportunity to apply the theory in an undergraduate course on Visual Literacy. This experience will provide the author the first opportunity to begin asking the questions outlined in this chapter, and the first opportunity outside the context of this study to begin improving Learning Object Design and Sequencing Theory. Finally, NSF has funded a research project that will support the study and improvement of LODAS through funding of the development of LODAS-supporting technology. Integration of LODAS principles into the previously mentioned NEEDS and SMETE digital libraries at U. C. Berkeley will also extend the theory’s reach and potential for improvement. As Eric Raymond has said of software, so may it be said of instructional design theory: “Given enough eyes, all bugs are shallow” (Raymond, 1999).

Implications for the Field. While the entirety of LODAS has the potential to impact practice in the field, as it is the first to explicitly support IEEE 1484.12 learning objects, its greatest potential for impact is in its taxonomy and role as an example or model for other instructional designers. LODAS can be broken into three major sections: instructional design guidelines, learning object design guidelines, and “prescriptive linking material” that connects the instructional design guidelines to the learning object design guidelines. LODAS’ greatest potential for impact in the field of instructional technology is the example it sets in connecting instructional design theory to learning object design. Previously, any person or organization who wanted to employ learning objects in their instructional design and delivery was required to create their own taxonomy of learning object types. The author considers this to be a major cause of the current poverty of practical applications of the technology. However, taking the general taxonomy and learning object design guidelines presented in LODAS, any instructional designer may connect these to the instructional design theory of their choice via the creation of “prescriptive linking material,” a considerably simpler exercise than the creation of a new taxonomy. This development has the potential to (1) speed the practical adoption of the learning object approach, (2) allow the simplified application of any instructional design theory to the learning object approach, and (3) provide a common ground for future research of the instructional technology called “learning objects.”

REFERENCES

ADL. (2000). *Advanced distributed learning network website* [On-line]. Available: http://www.adlnet.org/

ALI. (2000). *Apple learning interchange website* [On-line]. Available: http://ali.apple.com/

Anderson, J. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. (2000). *ACT-R web based environment* [On-line]. Available: http://128.2.248.57/inter/ACT-R-tutorial.html

Anderson, T. & Merrill, M. D. (1999). *A design for standards-based knowledge components*. Unpublished manuscript.

ARIADNE. (2000). *Alliance of remote instructional authoring and distribution networks for Europe website* [On-line]. Available: http://ariadne.unil.ch/

Asymetrix. (2000). *Customer case study: Veturist, Inc*. [On-line]. Available: http://www.asymetrix.com/solutions/casestudies/venturist.html

Ausubel, D. P. (1968). *Educational psychology: A cognitive view*. New York: Holt, Rinehart, and Winston.

Bloom, B. S. (1956). *Taxonomy of educational objectives, handbook 1: Cognitive domain*. New York: Longmans Green.

Boehm, B. (1981). *Software engineering economics*. Englewood Cliffs, New Jersey: Prentice-Hall.

Brooks, F. P. (1995). *The mythical man-month: Essays on software engineering*. Reading, MA: Addison-Wesley.

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions. *The Journal of the Learning Sciences*, 2, 137-178.

Bruner, J. S. (1960). *The process of education*. New York: Random House.

Bunderson, C. V. (2000, April). *Design experiments, design science, and the philosophy of measured realism: Philosophical foundations of design experiments*. Symposium session at the annual meeting of the American Educational Research Association, New Orleans, LA.

Bunderson, C.V., Gibbons, A.S., Olsen, J.B., and Kearsley, G.P. (1981). Work models: Beyond instructional objectives. *Instructional Science, 10*, 205-215.

Bunderson, C. V., Martinez, M., & Wiley, D. (2000, April). *Verification in a design experiment context: Validity argument as design process*. Symposium session at the annual meeting of the American Educational Research Association, New Orleans, LA. Available: http://edumetrics.org/research/presentations/aera2000-wiley01.pdf

Bunderson, C. V., Newby, V., & Wiley, D. A. (2000.) *Invariant scaling and domain theories: A new foundation for design experiments*. (Manuscript in preparation).

Burns, H. L. & Capps, C. G. (1988). *Foundations of intelligent tutoring systems*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Burton, R. R., Brown, J. S., & Fischer, G. (1984). Skiing as a model of instruction. In B. Rogoff & J. Lave (Eds.), *Everyday cognition: Its development in social context* (pp. 139-150). Cambridge MA: Harvard University Press.

Carlson, R. A., Khoo, H., & Elliot, R. E. (1990). Component practice and exposure to a problem solving context. *Human Factors, 31*, 489-502.

Chen, M. (2000). *The history and paradigm shifts in educational computing* [On-line]. Available: http://www.telelearning.education.mcgill.ca/chen/c2notes.html

Clark, D. (1997). *Instructional system design - development phase* [On-line]. Available: http://www.nwlink.com/~donclark/hrd/sat4.html

Collins, A. (1990). *Toward a design science of education*. Cambridge, MA: Bolt Beranek and Newman.

Cronbach, L. J. (1988). Five perspectives on validity argument. In H. Wainer & H. I. Braun (Eds.), *Test Validity* (pp. 3-17). Hillsdale, NJ: Lawrence Erlbaum.

Dahl, O. J. & Nygaard, K. (1966). SIMULA - An algol based simulation language. *Communications of the ACM*, *9* (9), p. 671-678.

DCMI. (2000). *Dublin core metadata initiative education working group website* [On-line]. Available: http://purl.org/DC/groups/education.htm

Deming, W. E. (1995). *The new economics: For industry, government, education* (2nd ed.). Cambridge, MA: MIT Press.

Dijkstra, S., Seel, N., Schott, F., & Tennyson, R. (Eds.) (1997). *Instructional design: International perspectives*. Mahwah, NJ: Erlbaum.

EOE. (2000). *Educational objects economy website* [On-line]. Available: http://www.eoe.org/eoe.htm

ESCOT. (2000). *Educational software components of tomorrow website* [On-line]. Available: http://www.escot.org/

Forte, E., & Resmer, M. (1998). *ARIADNE - IMS agreement* [On-line]. Available: http://ariadne.unil.ch/Metadata/agreement.html

Gagne, R., Briggs, L. & Wager, W. (1992). *Principles of instructional design* (4th Ed.). Fort Worth, TX: HBJ College.

Gibbons, A.S., Bunderson, C.V., Olsen, J.B., and Rogers, J. (1995). Work models: Still beyond instructional objectives. *Machine-Mediated Learning, 5*(3&4), 221-236.

Gibbons, A. S. & Fairweather, P. G. (1998). *Computer-based instruction: design and development*. Englewood Cliffs, NJ: Educational Technology Publications.

Gibbons, A. S., Nelson, J., & Richards, R. (in press). The nature and origin of instructional objects. In D. A. Wiley (Ed.), *The instructional use of learning objects*. Bloomington, IN: Association for Educational Communications and Technology.

Greeno, J. G., Collins, A., Beranek, B., & Resnick, L. B. (1994). Cognition and Learning. In D. Berliner & R. Calfee (Eds.), *Handbook of educational psychology*. 1-51.

Heinich, R., Molenda, M., Russell, J. D., and Smaldino, S. (1999). *Instructional media and technologies for learning*. Sixth Edition. Englewood Cliffs, NJ: Prentice-Hall.

Hodgins, Wayne. (2000). *Into the future* [On-line]. Available: http://www.learnativity.com/download/MP7.PDF

IMS. (2000a). *Instructional management systems project website* [On-line]. Available: http://imsproject.org/

IMS. (2000b). *Instructional management systems project website* [On-line]. Available: http://imsproject.org/imMembers.html

Internet Newsroom (1999). *Internet growing too fast for search engines* [On-line]. Available: http://www.editors-service.com/articlearchive/search99.html

L’Allier, J. J. (1998). *NETg's precision skilling: The linking of occupational skills descriptors to training interventions* [On-line]. Available: http://www.netg.com/research/pskillpaper.htm

LOM (2000). *LOM working draft v4.1* [On-line]. Available: http://ltsc.ieee.org/doc/wg12/LOMv4.1.htm

LTSC. (2000). *Learning technology standards committee website* [On-line]. Available: http://ltsc.ieee.org/

Marciniak, J. J. (Ed.). (1994). *Encyclopedia of software engineering*. New York, NY: Wiley & Sons.

Martinez, M. (1999). *An investigation into successful learning--Measuring the impact of learning orientation, a primary learner-difference variable, on learning*. Doctoral dissertation, Brigham Young University.

McCracken, D. D. (1981). A maverick approach to systems analysis and design. In W. Cotterman, et al., (Eds.), *Systems analysis and design: A foundation for the 1980’s*. Amsterdam, The Netherlands: North-Holland.

MERLOT. (2000). *Multimedia educational resource for learning and on-line teaching website* [On-line]. Available: http://www.merlot.org/

Merrill, M.D. (1983). Component display theory. In C. Reigeluth (ed.), *Instructional design theories and models*. Hillsdale, NJ: Erlbaum Associates.

Merrill, M.D., Li, Z. & Jones, M. (1991). Instructional transaction theory: An introduction. *Educational Technology, 31*(6), 7-12.

Merrill, M. D. (1999.) *Top of the top 10 in instructional design* [On-line]. Available: http://www.amazon.com/exec/obidos/ASIN/0877782989/o/qid=953680365/sr=8-1/103-4305913-5208626

Merrill, P. F. (1976). Task analysis: An information processing approach. *NSPI Journal, 15*(2), 7-11.

Messick, S. (1998). *Consequences of test interpretation and use: The fusion of validity and values in psychological assessment*. Princeton, NJ: Educational Testing Service, RR-98-4.

Microsoft. (2000). *Resources: Learning resource interchange* [On-line]. Available: http://www.microsoft.com/eLearn/resources/LRN/

Nelson, L. M. (1998). *Collaborative problem solving: An instructional theory for learning through small group interaction*. Unpublished doctoral dissertation, Indiana University.

Niemiec, R. P., & Herbert, J. W. (1989). From teaching machines to microcomputers: Some milestones in the history of computer-based instruction. *Journal of research on computing in education. 21*(3), 263-276.

Paas, F.G.W.C., & van Merriënboer, J. J. G. (1992). An instructional design model for the training of complex cognitive skills. *Tijdschrift voor Onderwijsresearch*, 17, 17-27.

Paas, F.G.W.C., & van Merriënboer, J. J. G. (1994a). Variability of workload examples and transfer of geometrical problem solving skills: A cognitive load approach. *Journal of Educational Psychology, 86*, 122-133.

Paas, F.G.W.C., & van Merriënboer, J. J. G. (1994b). Measurement of cognitive load in instructional research. *Educational Psychology Review*, *6*, 351-371.

Pagliaro, L.A. (1983). The history and development of CAI: 1926-1981, an overview. *The Alberta Journal of Educational Research*, *29*(1), 75-84.

Pask, G. (1975). *Conversation, cognition, and learning*. New York: Elsevier.

Patterson, C. (1977). *Foundations for a theory of instruction and educational psychology*. New York: Harper & Row.

Pressman, R. S. (1992). *Software engineering: A practitioner's approach*. McGraw-Hill, New York.

Pressey, S. L. (1927). A machine for automatic teaching of drill material. *School and Society*, 25, 549-52.

Raymond, E. (1999). *The cathedral & the bazaar: Musings on Linux and Open Source by an accidental revolutionary*. Monterey, CA: O'Reilly.

Recker, M., & Wiley, D. A. (2000). *A non-authoritative educational metadata ontology for the collaborative filtering and recommendation of learning objects*. (Manuscript in preparation).

Reeves, T. C. (1993). Pseudoscience in computer-based instruction: The case of learner control research. *Journal of Computer-Based Instruction*, *20*(2), 39-46.

Reigeluth, C.M. (1979). In search of a better way to organize instruction: The elaboration theory. *Journal of Instructional Development*, *2*(3), 8-15.

Reigeluth, C.M., Merrill, M.D., Wilson, B.G., & Spiller, R.T. (1980). The elaboration theory of instruction: A model for structuring instruction. *Instructional Science*, *9*(3), 195-219.

Reigeluth, C.M., & Darwazeh, A.N. (1982). The elaboration theory's procedure for designing instruction: A conceptual approach. *Journal of Instructional Development, 5*(3), 22-32.

Reigeluth, C. M. (1983). Instructional design: What is it and why is it? In C. M. Reigeluth (Ed.), *Instructional design theories and models: An overview of their current status* (pp. 3-36). Hillsdale, NJ: Lawrence Erlbaum Associates.

Reigeluth, C.M. (1992). Elaborating the elaboration theory. *Educational Technology Research and Development, 40*(3), 80-86.

Reigeluth, C. M. & Nelson, L. M. (1997). A new paradigm of ISD? In R. C. Branch & B. B. Minor (Eds.), *Educational media and technology yearbook* (Vol. 22, pp. 24-35). Englewood, CO: Libraries Unlimited.

Reigeluth, C. M. (1999a). The elaboration theory: Guidance for scope and sequence decisions. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory*. (pp. 5-29). Hillsdale, NJ: Lawrence Erlbaum Associates.

Reigeluth, C. M. (1999b). What is instructional design theory and how is it changing? In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory*. (pp. 5-29). Hillsdale, NJ: Lawrence Erlbaum Associates.

Reigeluth, C. M. & Frick, T. W. (1999). Formative research: A methodology for creating and improving design theories. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory*. (pp. 5-29). Hillsdale, NJ: Lawrence Erlbaum Associates.

Richey, R. C. (1986). *The theoretical and conceptual bases of instructional design*. London: Kogan Page.

Rowland, G. (1992). What do instructional designers actually do? An initial investigation of expert practice. *Performance Improvement Quarterly, 5*(2), 65-86.

Royce, W.W. (1970). *Managing the development of large software systems: Concepts and techniques*. Proceedings of the Nineteenth Annual IEEE WESCON Convention.

Sack, W., Soloway, E., & Weingrad, P. (1994). Re-writing cartesian student models. In J. E. Greer & G. I. McCalla (Eds.), *Student modelling: The key to individualized knowledge-based instruction*. (NATO ASI Series ed., Vol. 125, pp. 355-376). Berlin: Springer-Verlag.

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999). Learning by doing. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory*. (pp. 161-181). Hillsdale, NJ: Lawrence Erlbaum Associates.

Shute, V. J., & Psotka, J. (1994). Intelligent tutoring systems: past, present, and future. In D. Jonassen (Ed.), *Handbook of research on educational communications and technology* (pp. 570-600). Scholastics Publications.

Simon, H. A. (1969). *Sciences of the artificial*. Cambridge, MA: MIT Press.

Skinner, B. F. (1958). Teaching machines. *Science*, 128, 969-77.

Snelbecker, G. E. (1974). *Learning theory, instructional theory, and psychoeducational theory*. New York: McGraw-Hill.

Snow, R. E. (1971). Theory construction for research on teaching. In R. M. W. Travers (Ed.), *Second handbook of research on teaching*. Chicago: Rand McNally.

Strauss, A. & Corbin, J. (1998). *Basics of qualitative research*. Thousand Oaks, CA: Sage.

Strong-Krause, D. (2000, April). *Developing theory-anchored, sample and task-set invariant measurement scales in spoken english as a second language*. Symposium paper at the annual meeting of the American Educational Research Association, New Orleans, LA.

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

Szabo, M. & Fuchs, A. (1999). *The IBM 1500 system: Historical perspectives* [On-line]. Available: http://www.quasar.ualberta.ca/EDIT572/572h3.html

Tennyson, R., Schott, F., Seel, N., & Dijkstra, S. (Eds.) (1997). *Instructional design: International perspectives*. Mahwah, NJ: Erlbaum.

Tuttle, J. G. (1970). *The historical development of computer capabilities which permitted the use of the computer as an educational medium in the United States from 1958 to 1968, with implications of trends*. Unpublished doctoral dissertation, New York University.

Urdan, T. A. & Weggen, C. C. (2000). *Corporate e-learning: Exploring a new frontier* [On-line]. Available: http://wrhambrecht.com/research/coverage/elearning/ir/ir_explore.pdf

van Merriënboer, J. J. G. (1990a). Strategies for programming instruction in high school: Program completion versus program generation. *Journal of Educational Computing Research, 6*, 265-287.

van Merriënboer, J. J. G. (1990b). *Teaching introductory computer programming: A perspective from instructional technology*. Enschede, The Netherlands: Bijlstra & van Merriënboer.

van Merriënboer, J. J. G. & De Croock, M. B. M. (1992). Strategies for computer-based programming instruction: Program completion versus program generation. *Journal of Educational Computing Research, 8*, 365-394.

van Merriënboer, J. J. G. ; Jelsma, O.; Paas, F.G.W.C. (1992.) Training for reflective expertise: A four-component instructional design model for complex cognitive skills. *Educational Technology Research and Development*, *40*(2), pp. 23-43.

van Merriënboer, J. J. G., & Dijkstra, S. (1996). The four-component design instructional design model for training complex cognitive skills. In R.D. Tennyson, F. Schott, N. Seel, and S, Dijkstra, (Eds), *Instructional design: international perspectives*. Mahwah, NJ: Lawrence Erlbaum Associates.

van Merriënboer, J. J. G., (1997). *Training complex cognitive skills: A four-component instructional design model for technical training*. Englewood Cliffs, NJ: Educational Technology Publications.

WebCT. (2000a.) *WebCT background* [On-line]. Available: http://about.webct.com/company/index.html

WebCT. (2000b.) *WebCT network learning environment* [On-line]. Available: http://about.webct.com/company/env.html

Wiley, D. A. (in press). An introduction to learning objects. In D. A. Wiley (Ed.), *The instructional use of learning objects.* Bloomington, IN: Association for Educational Communications and Technology.

Wiley, D. A., South, J. B., Bassett, J., Nelson, L. M., Seawright, L. L., Peterson, T., & Monson, D. W. (1999). Three common properties of efficient online instructional support systems. *The ALN Magazine, 3*(2), [On-line]. Available: http://www.aln.org/alnweb/magazine/Vol3_issue2/wiley.htm

Wilson, B., & Cole, P. (1992). A critical review of elaboration theory. *Educational Technology Research and Development, 40*(3), 63-79.