2010-12-02

Evolving Information Technology: A Case Study of the Effects of Constant Change on Information Technology Instructional Design Architecture

C. Richard G. Helps
Brigham Young University - Provo

Evolving Information Technology: A Case Study of the Effects of Constant Change on Information Technology Instructional Design Architecture

Richard Helps

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

Andrew S. Gibbons, Chair
Randall S. Davies
Charles Graham
Paul F. Merrill
David D. Williams

Department of Instructional Psychology and Technology

Brigham Young University

November 2010

ABSTRACT

Evolving Information Technology: A Case Study of the Effects of Constant Change on Information Technology Instructional Design Architecture

Richard Helps

Department of Instructional Psychology and Technology

Doctor of Philosophy

A major challenge for Information Technology (IT) programs is that the rapid pace of evolution of computing technology leads to frequent redesign of IT courses. The problem is exacerbated by several factors. Firstly, the changing technology is the subject matter of the discipline and is also frequently used to support instruction; secondly, this discipline has only been formalized as a four-year university program within recent years and there is a lack of established textbooks and curriculum models; finally, updating courses is seldom rewarded in a higher education system that favors research and teaching for promotion and tenure. Thus, continuously updating their courses place a significant burden on the faculty.

A case study approach was used to describe and explain the change processes in updating IT courses. Several faculty members at two institutions were interviewed and course changes were identified and analyzed. The analysis revealed a set of recurrent themes in change processes. An instructional design architecture approach also revealed a set of design domains representing the structure of the change processes. The design domains were analyzed in terms of the design decisions they represented, and also in terms of structures, functions and activities, which are related to Structures-Behaviors-Functions (SBF) analysis.

The design domains model helped to explain both negative and positive outcomes that were observed in the data. When design efforts impact multiple domains the design is likely to be more difficult. Understanding the design domain architecture will assist future designers in this discipline.

Keywords: Instructional Design Architecture, Instructional Design Layers, Structures Behaviors Functions Analysis, Design decisions, Design Domains, Evolving Technology, Information Technology Course Design, Rate of Change of Information Technology, Instructional Design Theory

ACKNOWLEDGMENTS

Reading maketh a full man, conference a ready man, and writing an exact man

- Francis Bacon

I am grateful to all those who influenced my life during this learning phase. Through this experience I learned to read the source documents, not someone else's opinion of them (thanks colleagues and fellow graduate students—you taught me that!); then to interact with people who have given much of their lives to thinking about and doing those things—my opinions changed (thanks committee and colleagues!); then to write down my understanding and have someone else critique it—I learned lots from that process too—then rewrite it (thanks to all my editors, formal and informal!).

Thanks to the faculty, staff, family and friends who were always supportive and positive throughout this process. Thanks to my committee chair, Dr. Andy Gibbons, who was unfailingly positive and supportive in encouraging me with this research. I must also say a special word of thanks to my wife and family who not only provided moral support but also donated many hours to the details of organizing and writing this research project.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction and Statement of the Problem

Information technology is a rapidly changing field encompassing many disciplines related to computing. In the past decade a university-level academic discipline, also somewhat ambiguously called Information Technology (IT), has emerged and is growing as fast as its parent technology field. The academic discipline of IT faces a number of challenges, including a shortage of educational academic resources; a recently emerged program definition; a very rapidly changing, complex technical discipline; instructional designers with little training in instructional design; and an academic system that fails to reward ongoing and time consuming efforts to keep instructors and curriculum up to date.

Firstly, IT is a relatively new discipline at the university level; four year IT degree programs at American universities emerged from related Computer Science, Information Systems, and Electronics and Computer Engineering or Technology programs, starting about 1995. Standards for a four-year IT curriculum, consisting of lists of technical topics, were proposed to the ACM and IEEE Computer Society professional organizations in 2005 and accepted in final form in 2008 (Lunt , et al., 2008). Formal standards for accrediting four-year IT programs were negotiated with ABET over several years and finally approved for evaluating programs in 2007 (ABET, 2007).

IT programs distinguish themselves from related disciplines by focusing on integration of computing sub-systems rather than on hardware or software design. Papers have been published debating the essential nature of IT education (Ekstrom & Lunt 2003; Lunt , et al., 2008; Lunt, Lawson, Goodman, & Helps, 2002). The vast majority of this foundational work is focused on the technical domain content of the curriculum, with some discussions on strategies for teaching. Much less work has been done in studying the instructional architecture and how curricula in this field evolve as changes occur in the field. The net effect of this pattern of development is that instructors at various institutions are pragmatically developing and modifying classes at a rapid rate, either using textbooks from related but different disciplines or generating their own materials, while they continue to investigate new technologies for their own professional interest and incorporate them into their teaching.

Secondly, both because it is a professional degree program and because it follows the tradition of other technical technology degree programs, IT instruction emphasizes and champions authentic instruction using current technology (Gorka, Miller, & Howe, 2007; Lamancusa, Zayas, Soyster, Morell, & Jorgensen, 2008). Many technology programs use phrases like “hands-on” or “experiential” to characterize their educational approach (Gorka, et al., 2007; Jamieson, Edwardson, & Lohmann, 2009; Kolb, 1984; Lindsey & Berger, 2009). This instructional approach differentiates technology programs from science programs and even from many engineering programs. For IT students to graduate as competent professionals in their discipline, it is necessary for them to be able to understand and apply current technological developments. Since the computer technology used in this experiential education changes constantly and rapidly (Brock & Moore, 2006; Molebash, 2000; Mollick, 2006; Moore, 1965; Roberts, 2000; Schaller, 1997), as opposed to the relative stability of the underlying concepts and theories, the student learning experiences need to be constantly updated.

Thirdly, IT instructors, whose educational background emphasizes technology rather than instructional design, usually design the courses. Their having little or no formal training in instructional design impacts their approach to design of instruction: they emphasize technical topics in their instructional design. It appears that the change process is a natural and on-going one, with little or no conscious consideration of the impact of on-going change on the structure of the designed educational experiences. Lidtke (1998) comments,

> The process of curriculum design in the computing sciences has changed little in the last three decades, while the content of computing curricula has changed continuously because of the technology upon which it is based and the explosive growth of knowledge in the field. (p. 954)

As further evidence of the relative emphasis on technical advances in preference to theoretical educational design considerations in this domain, Lister and Box (2008) recently completed an analysis of papers published in the proceedings of the Computer Science Education Special Interest Group (SIGCSE) annual conference. They concluded, "The epistemology of the SIGCSE community is primarily objectivist, with a focus on content, rather than a constructivist, student-centered focus on learning." A similar informal analysis of papers published in the Special Interest Group for Information Technology Education of the ACM (ACM-SIGITE) conference proceedings supports this conclusion, with the majority of the published papers focusing on new technology developments that could be introduced into the classroom, with little or no consideration of the educational curriculum architecture required or affected by these changes (*Proceedings of the 9th ACM SIGITE conference*, 2008).

Fourthly, faculty tenure and promotion decisions in computing disciplines are commonly based on successful research and peer-reviewed publication; upgrading curricula is seldom well recognized and rewarded for faculty in higher education academic environments. Thus, in a rapidly changing technical discipline, updating their own knowledge as well as updating their constantly changing curricula places significant stresses on busy faculty members with little reward other than meeting the needs of their students.

The combination of these competing forces—a developing discipline with an emphasis on current technology, a constantly and rapidly changing set of current technologies, and a lack of reward for curricular re-design strongly suggests that the impact of instructional design for instructors needs to be minimized.

While it is apparent that curricula are evolving rapidly, the nature of the changes is not as clear. A clear understanding of what and how changes occur is a necessary prerequisite to developing better paradigms for course design for IT. The problem thus posed is to attempt to explore what is happening as IT curricula evolve. What happens to the instructional design as curricula are updated? What are instructors and designers doing and how are they doing it? This deeper understanding of what is happening will create a foundation upon which a methodology for evolving curricula in rapidly changing environments may be developed. This study is an exploration of the change process.

In exploring the topic of existing curricular change an appropriate theoretical stance must be selected, whether it be objectivist or naturalistic or some other approach. Guba and Lincoln (1989), in their discussion of constructivist research paradigms, tell us that any exploration and discovery will be theory-laden and that we do better by admitting and embracing the theory that defines the viewpoint of the research, rather than claiming an indefensible purely objective viewpoint (pp. 57-68). Since the goal is to study an existing process of change a naturalistic approach is suggested. In addition, using the paradigm of instructional design architecture is helpful in discussing curricular change effects. If the change process can be examined and interpreted in terms of effects of change on the structure of the design then we gain a nuanced and meaningful view of what is happening and can interpret it in terms of educational theory. This research project uses a case-study approach to examine a small cross-section of curricular changes in the IT domain. The results are interpreted and described through an architectural view of instructional design. The goal is to illustrate the changes in terms of the underlying instructional design structures.

This research project will explore how instructional design evolves under technological change and other influences, and describe the nature of the changes that occur.

Research Questions

On the basis of the above introduction there are two research questions addressed by this study. Each research question is defined by several sub-questions. The questions address the natures of the changes that take place as courses are re-designed and the second question addresses how instructors, who are the course designers, interact with the process of change.

The first question is, what is the nature of the changes that occur in instructional design architecture when courses in higher-education information technology evolve? This question has six sub-questions.

- What effect do changes have on the instructional design architecture?
- How frequently do changes occur?
- What are the types of changes that occur (content, procedures, equipment)?
- What is the range of extent and granularity of the changes?
- What motivates the changes?
- How are the changes managed i.e. what models or templates (deliberate or unconscious) are used to manage the change process?

The second question relates to how the instructors interact with the process of change. The question is, what actions or processes do instructors implement when changing courses? This question has five sub-questions.

- What actions do instructors take in order to keep to keep their workload manageable?
- To what extent is curriculum change a solo effort and to what extent is it shared or delegated?
- What do instructors do to ensure the students' educational experience isn't harmed by updating the curriculum?
- What principles do course designers/instructors use in deciding what to change and what to keep?
- What choices (trade-offs) do instructors make when modifying the curriculum?

This research analyzed evolving higher education Information Technology course design. Using a case-study approach the researcher describes the changes that occur in higher education IT and addressed the questions posed. The changes were interpreted in terms of design architecture effects. Development of new design methods is not included in this research project.

Glossary of Terms

Several terms from a variety of sources will be used throughout this dissertation. They are defined as follows:

Constructivism. A theory of learning based on the premise that knowledge is constructed in the mind of the learner, rather than having an objective reality that is discovered by the learner.

Naturalistic Inquiry. Naturalistic inquiry is inquiry into phenomena in their natural setting, rather than in laboratory or contrived settings. Naturalistic inquiry uses a disciplined approach and inquiry standards to perform rigorous research. Naturalistic inquiry has close ties to other forms of research, such as ethnography and qualitative research.

Qualitative research methods. Qualitative methods use naturalistic approaches to explore phenomena within their context, accepting that the researcher also impacts the results obtained. Qualitative research methods are usually contrasted with quantitative methods, which rely heavily on statistical analysis and statistical requirements of objectivity, independence and experimental control.

Generalizable vs. transferable. Research that can be formulated into a set of rules or formulae and applied situations other than the specific research where they were derived, is generalizable. This is characteristic of (valid) quantitative research. Research that can be understood and interpreted as necessary by readers and then applied to their own circumstances, is transferable This is characteristic of (disciplined) naturalistic or qualitative inquiry.

Thick or rich description. As applied to case study research, thick description and its companion, thick interpretation, include the case's own issues, contexts and interpretations. Rather than being pared down to the essential minimum amount of information thick descriptions include enough of the background and emic information to aid the reader in understanding the context of the results. Thick descriptions are expected to include multiple, even conflicting, presentations of the findings. (Stake, 2000)

Emic vs. etic perspectives. The inquirer's perspective or values in interpreting findings or observations is referred to as the etic perspective. The participant's perspective and values are referred to as the emic perspective. (Vidich & Lyman, 2000, p. 41)

Structures, Behaviors and Functions (SBF). Structures, Behaviors and Functions (SBF) or Function-Structure-Behavior (FSB) or several other similar variants all describe an analytical method for understanding designs used by multiple researchers and designers in the computing and instructional design fields. Aspects of the design are characterized in terms of each of the SBF properties

ABET Inc. or ABET. ABET is the largest and most widely recognized body for accrediting engineering, technology, computing and applied science programs. The acronym used to stand for the Accreditation Board for Engineering and Technology but they formally changed their name to "ABET Inc." and declared that it is no longer an acronym, since they now accredit many programs that are neither engineering nor technology.

Procedural programming. Procedural programming is a style of computer language where the program is written as a sequence of instructions that the computer executes. This style was popular for many years and is the basis for computer languages such as Fortran, COBOL, C, Pascal and many others. Procedural programming has been largely replaced by the newer object-oriented programming

Object-Oriented Programming (OOP). Object-oriented programming is a style of computer language where, instead of writing a sequence of instructions, a program is written by creating objects, each representing some entity. Each object has its own data and methods. Procedural programming has been likened to a series of verbs commanding the computer and OOP has been likened to adding nouns and adjectives to those verbs. OOP is currently the most popular style of programming and is used in computer languages like Java, Python, C++, Ruby and Smalltalk and many others.

Chapter 2: Review of the Literature

Three bodies of educational research literature have particular relevance to the topics addressed in the research questions. These inform and guide the research study. They are curriculum theory, design of instruction within disciplines (specifically instruction within Information Technology (IT)), and instructional design architecture theory as part of design theory.

Curriculum theory provides a background for understanding curriculum structure. Instruction design within disciplines provides insights into the technology-specific applications of instructional design. Instructional design architecture provides a conceptual paradigm for understanding and interpreting the research results.

Each of these three aspects will be discussed and their effect on the research study considered.

Curriculum Theory

A curriculum has been defined as “All the learning which is planned and guided by the school,” (Kelly, as cited in (Smith, 1996, 2000)). Smith further breaks down this very broad description of curriculum: “four ways of approaching curriculum theory and practice:

1. Curriculum as a body of knowledge to be **transmitted**.
2. Curriculum as an attempt to achieve certain ends in students—**product**.
3. Curriculum as **process**.
4. Curriculum as **praxis**.” (Smith, 1996, 2000, p. 2)

Smith’s first category, curriculum as a body of knowledge, incorporates the assumption that a curriculum is a series of topics or skills that the learner must assimilate. Designers in technical disciplines such as computing, easily and often unconsciously accept this as the definition of curriculum. Textbooks list a series of technical topics in logical order; lesson plans often follow a similar structure. This approach to curriculum design resonates with the curriculum definitions proposed by the CC2001 and CC2005 models and widely cited by computing curricula designers (*Computing Curricula 2001: Computer Science*, 2001; *Computing Curricula 2005: The Overview Report*, 2005). These publications are the reports of the Computing Curricula Program, which is a joint program sponsored by a few of the leading computing professional societies to define the curricula for several different disciplines in computing higher education including computer science, information technology and information systems. Uniformly, these proposed curricula are organized around lists of technical topics plus lists of specific professional skills. For example, the most detailed information about the curriculum is contained in chapter 3 of the Computing Curricula 2005 report. Chapter 3 discusses the intended outcomes for graduates from computing programs. The key information in this chapter is contained in three tables (*Computing Curricula 2005: The Overview Report*, 2005, pp. 24, 25, 28)(Tables 3.1, 3.2, 3.3) that present in great detail tens of technical topics the students should address. There is no corresponding discussion of educational strategy or non-technical learning objectives. The closest the document comes to these issues is an indication of the relative emphasis on applied versus theoretical knowledge of the different disciplines. Computing departments use these model curricula in designing or validating their own curricula. Therefore, it is hardly surprising that the curricula thus created are strongly oriented towards technical topics, rather than any broader view of instructional design.

This topic-first view of curriculum design influenced the research study in that it at least partially described the curriculum model found in the case studies that were investigated. The technical topics approach to curriculum design has the attraction of being systematic and methodological. It subscribes to a logical organization of teaching objectives and topics and appeals to the logical and objectivist preference of the faculty in computing disciplines.

Smith's second definition of curriculum is curriculum as a product. Curriculum as a product takes the view that the purpose of a curriculum is to produce a graduate capable of achieving certain specific outcomes. Curriculum theorists such as Bobbit and Tyler have supported it historically (Tyler, 1949). The current trend in higher education towards an outcomes-oriented approach has been enthusiastically adopted by the computing education community. Most programs in IT and Computer Science are or will be accredited by ABET, which is strongly committed to this model of educational design (ABET, 2008). The limitation of this view of curriculum design is that there is a degree of uncertainty about what is being measured. It is possible, although not necessarily easy, to state clear learning outcomes and to measure those outcomes. However, the causal versus correlational links between educational practices and actual outcomes are less clear. Specific learning outcomes, translated into practical course design, may or may not lead to hoped-for but less specific learning goals. In addition, a focus on outcomes may ignore other valuable learning that may occur; learning that is not expressed in terms of a specific outcome may be neglected. This places a strong responsibility on developers to include all desired learning in terms of specific outcomes, not just technical-topic outcomes. Given the current state of model curricula in computing it is likely that some valuable learning outcomes will be omitted. Since many IT programs and all three of those studied used outcomes as discussed here, this too described issues relevant for this study. The relationship between specified outcomes and actual learning was an issue, as is discussed later in this report.

A third view of curriculum design is that of "curriculum as a process." This viewpoint considers classroom interactions as part of the designed educational experience. The objectives of the curriculum are viewed as an intention, but the process is viewed as what actually happens with interactions between learner and instructor. Stenhouse (1975) describes this two-part learning design as follows, “A curriculum, like the recipe for a dish, is first imagined as a possibility, then as the subject of experiment.” And further, “Finally, within limits, a recipe can vary according to taste. So can a curriculum.” (pp. 4, 5) This viewpoint informs the research study in that the intentions (objectives) need to be considered as well as the actions. In other words, the curriculum is designed with specific aims and objectives in mind but these may change during instructional delivery. In fact, following Stenhouse’s train of thought a little further, the curriculum may evolve away from its original design by instructors making changes as they go, just as an experienced chef may change a familiar recipe according to changing needs. This is certainly a factor of interest for this research study.

This viewpoint is also reflected by Gibbons and Helps (in preparation) which discusses the different design considerations observed during design and during delivery. This is of relevance to the current research in that the changes that take place in instruction may occur through a specific design activity or may be implemented by the instructor on a less formal basis, with or without consideration for what effect those changes may have on the instructional architecture. These changes of curriculum as executed can be evaluated by interviewing the instructor and by reviewing artifacts of curricular design.

These views of curriculum structure provided a foundation from which changes in curriculum were considered. Their reality within the scope of computing education influenced the kinds of results sought and found and also the kinds of variances from those expectations that indicated useful data about the educational structure and any changes in that structure that were observed.

Instruction Within the Discipline

Literature relating to design of instruction within IT is necessarily sparse, since the higher-education discipline of IT has only been officially recognized within the last decade. IT, however, has roots in Computer Engineering, Computer Technology and Computer Science. These are themselves relatively new technical disciplines but do have some five to seven decades of history as opposed to the single decade of IT. Educational design from those fields provides valuable insight. Computing curricular design in recent years often refers to the "Computing Curricula 2001" documents (*Computing Curricula 2001: Computer Science*, 2001). The documents originally focused on Computer Science and were subsequently extended to include curricula for Computer Science, Computer Engineering, Information Systems and Information Technology (*Computing Curricula 2005: The Overview Report*, 2005) as discussed earlier. Sections of this volume have been extended and revised, specifically the section for Information Technology (Lunt , et al., 2008). These standards address the content of computing curricula rather than instructional architecture. As discussed above, this defines curriculum as a body of knowledge. Many research publications refer to these documents and likewise address issues of content and strategy.

As discussed above there is evidence that instructional researchers within computing higher education frequently focus on technical content rather than on instructional design principles. In fact, there is evidence that instructional designers make little use of learning theories. In a study of the closely related field of engineering education, a recent report by the American Society of Engineering Education (ASEE) the authors commented (Jamieson, et al., 2009),

Seldom are engineering educational innovations grounded in confirmed learning theories

> and pedagogical practices (National Academy of Engineering, 2005; Pellegrino, 2006), and many innovations, once implemented, are not assessed for their effectiveness in achieving their stated objectives. The trial-and-error nature and focus on technical content and technological tools neither systematically ensure that our graduates have the kind of educational experiences needed for the future nor assure the innovations created are replicable in other learning environments. (p. 4)

This focus on technical content is further shown by other recent studies. An analysis of citations of papers was published in the proceedings of the Computer Science Education Conference (SIGCSE). SIGCSE is accepted as the leading conference venue for computer science educators. In their paper, Lister and Box (2008) analyze the citations of 122 papers published in the proceedings of a recent SIGCSE conference. They were looking for common themes and for an accepted theoretical instructional design body of knowledge. Lister and Box indicate that the majority of the papers published focus on new technology, rather than on instructional theory. A quotation from the abstract summarizes their findings.

> The SIGCSE 2007 authors cited a very large array of conferences, journals and books, but the majority are only cited within a single paper. There are only a very small set of journals and conferences cited frequently. Most books cited are concerned with technical information or are textbooks. Only 2% of books are concerned with computer science education and 23% with education in general. The picture that emerges from this citation analysis is that the SIGCSE community does not have a substantial core set of educational literature. Also, the epistemology of the SIGCSE community is primarily objectivist, with a focus on content, rather than a constructivist, student-centered focus on learning. (p.1)

There are, of course, authors who address other areas than technical content. For example, Machanick (2003) specifically discusses the separation of principles from content. However, addressing principles as Machanick does is just a more abstract way of viewing technical content, not instructional design architecture. Other authors address the methodology of curriculum design of computing disciplines (Lidtke, 1998) among other issues, but overall there is a distinct bias towards new technological innovation rather than instructional design architecture.

Modularization, Domains and Instructional Design Architecture

From engineering to architecture, from art to education, over the years and particularly in recent decades design principles have emerged which translate across many divergent fields of application. The concept of modular design, which overlaps with concepts of design domains and layers of design, is important to this study. The benefits of modularity and layered design architectures significantly reduce design effort and redesign effort in fields where they are already deployed. To the extent that they are applicable to instructional design problems, they offer a strong potential solution for the problems described in this study. The ideas, benefits and applications of these concepts are presented here.

One of the drivers of our rapid technological progress over the last two centuries is the concept of incremental advances, combining ideas and developments from diverse creators and fields to create new systems. The effectiveness of this compounded sharing of ideas significantly depends on standardization and modularity.

Standardization means that designers and builders can share parts—a bolt made in China will screw onto a nut in a German car being built in England. Modularization is a higher order of standardization. Not just individual components but complete sub-systems must work together and must be removable and replaceable with alternative, equivalent (presumably superior) modules. Just as the threads in the nut and the bolt must match, so too the interfaces between cooperating modules must match. Modules are independent but related entities that can be combined in multiple ways to make working systems. Modularization is found in most if not all fields of design but one of the fields of design to benefit from this concept with exemplary success is design of computer systems, both hardware and software. Personal computers can be built from interchangeable parts and sub-systems, and most modern software is developed by selecting and extending modules from software libraries. There are benefits of modularization in computing systems that are translatable to instructional design. I will outline some of the benefits and principles in several design fields and then relate them to instructional design architecture theory.

The benefits of modularity both in the designed system and in the process of design of complex computer systems are described by Baldwin and Clark (2004).

> From an engineering perspective, modularity does many things. First, it makes the complexity of the system manageable by providing an effective "division of cognitive labor." It also makes possible the graceful evolution of knowledge about the system (Garud & Kumaraswamy, 2007). In this way, modular systems are "cognitively economic." Second, modularity organizes and enables parallel work. Work on or in modules can go on simultaneously, hence the start-to-finish time needed to complete the job decreases. Thus modular systems are "temporally economic." Finally, modularity *in the design* of a complex system allows modules to be changed and improved over time without undercutting the functionality of the system as a whole. In this sense, as we indicated above, the modular design of a complex system is "tolerant of uncertainty" and

"welcomes experimentation" in the modules. (pp. 6,7)

The authors further state that, "By definition a modular architecture allows module designs to be improved over time without undercutting the functionality of the system as a whole" (Baldwin & Clark, 2004, p. 9). And again, referring to the concept of modularity within the design process (as opposed to within the manufacturing process or in the final product) they state that, "Modularity-in-design allows users or system integrators to mix and match of the best designs within each module category and to incorporate new and improved module designs as they become available." (Baldwin & Clark, 2004, p. 31)

Thus Baldwin and Clark claim that, provided the design approach is appropriately modular, we can evolve systems gracefully, develop them using parallel teams creating separable modules and incorporate new modules as necessary. In addition the system will be tolerant of uncertainty and will welcome experimentation. Furthermore, if it is possible to "mix and match" instructional design modules and to improve instructional modules "over time without undercutting the functionality of the system as a whole," then we have addressed many of the concerns instructional designers face in rapidly evolving content domains.

All of these benefits, proven in computer environments, are extremely promising for the issues of concern to this study. If IT courses can be designed with an appropriate modular structure, with sub-sections that need to be replaced encapsulated in modules, then course evolution and updating will be greatly simplified. It is necessary to identify the characteristics of systems and modularity to see how they might apply to the problems under investigation.

Modularity in design and relative independence of modules from each other so they can be independently created or replaced by the design from other parts has been pursued in multiple disciplines. Brooks talks about this relative to several disciplines in "The Design of Design"

(Brooks, 2010), and the idea of independent modules has been pursued for some time in instructional design, with efforts to define and develop reusable learning objects (Allert, Dhraief, & W., 2002; Douglas, 2001; McGreal, 2004; Merrill, 2000; Robson, 2002; Sosteric & Hesemeier, 2002; Wiley, 2000).

Of course, not all good designs are modular, in the sense that they are or can be created by teams of people or are changeable over time. A notable class of exceptions is those artifacts created by a craftsman or artist. A painting, sculpture or other *oeuvre* created by a craftsman or artist may often be a holistic entity designed and created by one individual. This is a most effective mode of development where a single artifact is conceived, designed and executed within the mind of a single individual and which will remain static over time. Secondly, very simple designs are often not modular. When systems become complex, either through having multiple diverse sub-sections or by having multiple changes over time, which is the temporal equivalence of multiple sub-systems, then new factors come into play and aspects of complex systems must be taken into account. This exception is significant because course designers sometimes use this style in designing courses.

Baldwin and Clark (2000) discuss creating complex systems and propose that successful change and progress in many fields are driven by our ability to design and create complex systems and artifacts. They point out that computers are very complex artifacts and that no one person can create a computer system complete in all its details. However, if the complex system can be divided up so that different people can create the separate parts, then a group of people together can employ diverse skills to create a complex artifact. They describe two critical points along the scale of designing complex systems as follows:

Two interesting points arise as we move from simple to complex: (1) the point at which

> an artifact can no longer be made by a single person; and (2) the point at which an artifact can no longer be comprehended by a single person. Crossing into the first region requires a division of labor; crossing into the second requires a division of the knowledge and effort that goes into creating a design. (Baldwin & Clark, 2000, p. 6)

Thus if a system is complex and multifaceted, requiring either diverse skills in its design or requiring modular replacement over time, then modularity in design is beneficial or even essential.

Software modularity has been developed to advanced and abstract levels. An outstanding example of this approach is the very influential Design Pattern approach to designing new computer applications, made famous by the 'Gang of Four' (Gamma, Helm, Johnson, & Vlissides, 1995). Software design patterns, which were strongly influenced by ideas of Christopher Alexander's architectural design patterns (Alexander, Ishikawa, & Silverstein, 1977), are pre-made frameworks of software which solve specific types of computing problems. If programmers define the problem appropriately, they can then select a tested solution with known performance characteristics. These 'solutions' are not complete working solutions but are sophisticated design patterns, which encapsulate and represent the design decisions for similar problems. It is an important feature of design patterns that the problem parameters must be carefully specified and must match those of the design pattern. How the design pattern will interface to the rest of the system must also be carefully defined and matched to the problem being solved. If a problem's conditions and interface parameters match those of the design pattern then the design pattern is an appropriate design solution to the problem. The design pattern can then be implemented in whatever context that it is needed; usually this means it will be programmed in the same computer language as the rest of the system (Shalloway & Trott, 2002). Perhaps more important than providing templates for instant solutions, design patterns help designers to think about solutions and solution architectures in appropriate ways. The inventors of design patterns (Gamma, Helm, Johnson, & Vlissides, 1993) describe the design thinking aspects of the using patterns this way:

> Design patterns play many roles in the object-oriented development process: they provide a common vocabulary for design, they reduce system complexity by naming and defining abstractions, they constitute a base of experience for building reusable software, and they act as building blocks from which more complex designs can be built. Design patterns can be considered reusable micro-architectures that contribute to an overall system architecture. (p. 406)

Design patterns are a sophisticated example of modular design. In all modular design some basic conditions must be met. Design for modularity requires that each module's functionality and its interface with other parts of the system must be clearly defined so that it can be independently designed and, if necessary, replaced. Clear dividing lines need to be drawn between the modules to be created by different people or designed at different times; otherwise the system loses its essential modularity. Fortunately, guidelines have been developed in different disciplines to achieve this separation of parts. Parnas, an early pioneer in modularization of software, presented workable principles of modular software in his 1979 paper (D. L. Parnas, 1979) but he provided a key principle in an earlier paper.

> It is almost always incorrect to begin the decomposition of a system into modules on the basis of a flowchart. We propose instead that one begins with a list of difficult design decisions or design decisions which are likely to change. Each module is then designed to hide such a decision from the others. (David L. Parnas, 1972)

Brooks echoes this concept where he talks about design in terms of a tree of decisions (Brooks, 2010, pp. 192, 193). The abstract idea of modules being independent from each other and hiding the design decision they encapsulate from other modules is what makes the ideas of modularity so powerful in an environment where parts of the system must change while the remainder of the system is, one hopes, maintained. An important follow-up question then becomes, how does the designer divide up the design space?

Hmelo-Silver and Pfeffer and other educational researchers indicate that complexity in instructional design can be understood in terms of a Structure-Behavior-Function (SBF) model (Hmelo-Silver & Pfeffer, 2004; Jacobson & Wilensky, 2006). Collins and Ferguson refer to this as “form and function” analysis or as “structures, functions and processes” (Collins & Ferguson, 1993). These researchers suggest that most novices view complex systems in terms of the structures of the system, what Hmelo-Silver and Pfeffer describe as, “perceptually available, static components of the system” (2004, p. 127). On the other hand, experts relate the structures to the functions of the system, the functions and structures to behaviors of parts of the system and specifically how all these interact, thus adding functions and behaviors to the novice viewpoint. The work of these educational researchers was built on previous work of computing theory researchers who used SBF and related ideas for analyzing existing computer software (Goel & Chandrasekaran, 1989; Goel, Rugaber, & Vattam, 2009). Structures thus represent relatively static and easily identifiable components; functions represent descriptions of desired ends, and behaviors represent the activities to achieve the functions and operate on the structures. This representation of the design of complex systems (including instructional designs) goes beyond modularity and proposes that system design can be considered as being composed of abstractions conceived as structures, functions, behaviors or other inter-related but separable domains. The process of separating the abstract sub-systems is not trivial, as Christopher Alexander discusses at length (Alexander, 1964), and thus it is useful to have frameworks to guide our thinking as we approach this research problem. Parnas' idea of encapsulating decisions of design and various researchers' representations of SBF analysis provide two such analytical frameworks.

Blaauw and Brooks use concepts similar to SBF to describe the design of computer architecture (1997). They describe computer design domains of architecture, implementation and realization. All of these domains are design domains; none of them are part of building the system. Architecture, in their terminology, describes the function or functionality that is available to the user (or programmer). Implementation concerns the methods used to achieve this function, and realization concerns the means to materialize the method (p. 3). These three domains correspond roughly to the structure, function and behavior domains mentioned earlier. Blaauw and Brooks add something else to this discussion of design domains. They suggest design questions that are addressed in each of these domains. The questions and concerns are shown in Table 1.

They also make the point that for computers to be able to share software standardization is required, and for standardization to be achieved computers must share common architectures. These design viewpoints not only enlarge our understanding of design domains but provide possible questions for finding domain-related items and a better appreciation of what is required for modularization to be successful.

Table 1

Domains of Computer Design (from Blaauw and Brooks, p4)

Design Domains	Questions	Concerns
Architecture Functional appearance (to the systems programmer)	What?	Function
Implementation Logical Structure (performs the architecture)	How?	Method
Realization Physical structure (embodies the implementation)	Which? Where? When?	Means

In other fields of design, abstract layers or domains of design have been presented with different models. Stewart Brand addressed this problem with his architectural layers. He was describing an issue similar in concept to that of instructional design in rapidly evolving technology disciplines. His concern was how buildings evolve over time and how sub-sections of buildings are replaced to meet changing needs, which, while they do not evolve as rapidly as computer systems, have the same problems of complexity and entangled design layers.

Stewart Brand (1994) expressed his concept of evolution as, "how buildings learn." He defined six conceptual layers of any structure that can be used to deconstruct building designs. His six layers are alliteratively named Site, Structure, Skin, Services, Space-Plan, and Stuff. See Figure 1.

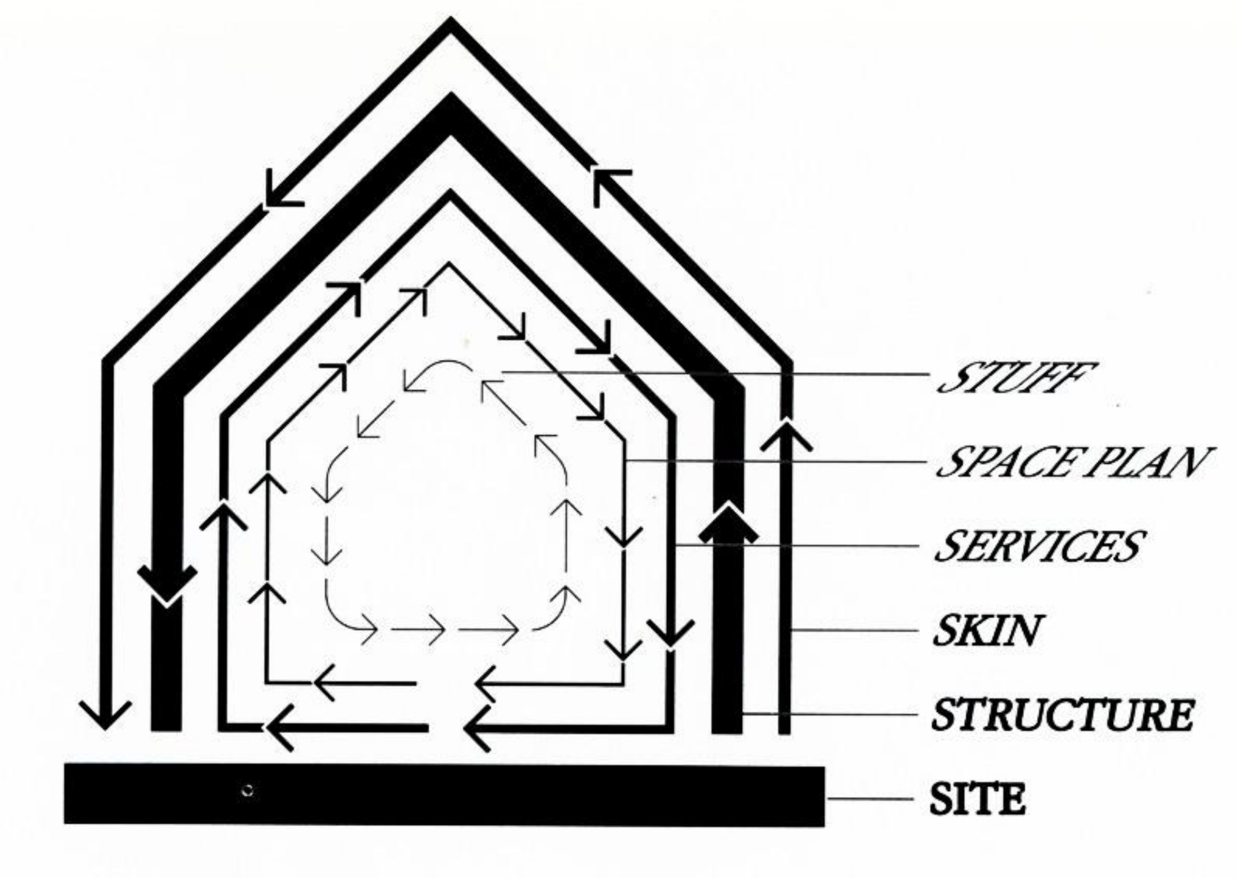

Figure 1. Stewart Brand's six layers of building which age at different rates (Brand, 1994, p. 13). The original caption stated, “SHEARING LAYERS OF CHANGE. Because of the different rates of change of its components, a building is always tearing itself apart.”

Brand describes how the different layers age at different rates and states that if a building is designed with an awareness of these layers then the building can evolve by changing the layers individually, without disturbing the rest of the structure. For example, it should be possible to change the electrical wiring (Services) of a building with minimal disruption to the other layers. If the electrical wiring is embedded in brickwork (Skin) then both layers will be impacted by a change. By way of comparison one can consider primitive buildings, such as mud huts, which are closer to the artistic artifacts mentioned earlier. In these systems the layers are compounded; for example the seating and working surfaces (Stuff) may well be inextricably combined with the walls (Structure), and the finish on the walls (Skin) is likewise part of the structure. The building is a holistic entity with intertwined layers. Any change in such a building has major implications for the other layers of the structure. Similarly, if a student learning experience is designed as a holistic entity with course content compounded with instructional strategy, class presentation materials and designed student interactions, then this course too will require major redesign if any of the entangled aspects of the course are changed.

One further example from the field of computer networks illustrates both the power of a layered or modular approach to design as well as some useful characteristics of practical application of these concepts. In computer networking a paradigm known as the ISO Open-Systems-Interconnect model (ISO-OSI) has been defined for analyzing and designing computer networks. A diagram of the model showing two users, represented by Host A and Host B, communicating through these conceptual layers is shown in Figure 2. The exact functioning of this system is not important to this discussion but a few characteristics are relevant. Firstly, this model is very widely accepted and is used as a discussion basis in all modern networking courses. Secondly, the users are not aware of the layers when using a computer. They interact with their computer through the Application Layer, the uppermost layer, Each of the other layers perform various operations on the message until the actual electrical signal is passed between the computers through the bottommost layer, the Physical Layer. For example, two computer users may exchange email and each one only interacts with their email software, the Application Layer, but all the other layers are effectively involved.

There are two important properties to this model. Firstly, every network between modern computers can be deconstructed using it. Each of the layers can be identified in a networking system through some overt or implicit function of the computers. Secondly, although this model is widely referenced both when designing and analyzing computer networks, it is not a design methodology. Some of the layers are often combined or split when building practical networks. A connection can be simplified or made more cheaply by combining the functionality of several layers in the design process.

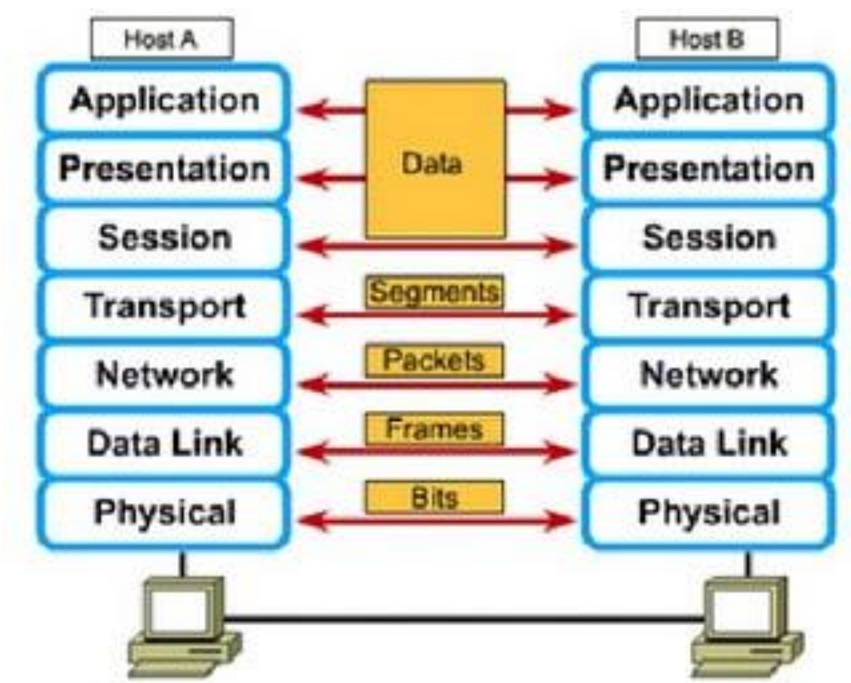

Figure 2. ISO_OSI Seven-layer networking model copied from http://purelycisco.blogspot.com

The lesson taken from this example for the current study is that conceptual layering of designs can be a powerful analytical tool but is not necessarily a methodology for design of practical systems.

An important model that must be introduced here is Schön's "Normative/Descriptive Design Domains" (Schön, 1987, pp. 58, 59) from the field of instructional design. In analyzing conversations of architects designing a school he defined a set of domains describing the design process. These domains are comparable to the layers or abstractions described previously and are also similar to the design layers model discussed below, in that each domain encapsulates a separate abstraction of one aspect of the design. Although Schön did not subdivide his domains there are different types of domain evident, such as the site and building elements (which the SBF model might identify as Structure); there are also his Program/use and Precedent domains that seem to reflect design intentions or functions, and there are other possible sub-categories (e.g. a meta-category called Explanation). A factor that is of considerable use to the current research is that these domains were not pre-chosen but emerged from an analysis of the conversations. This emergent structure or discovered theory approach fits into the qualitative analysis and grounded theory paradigms. Qualitative analysis will be discussed in more depth in the next chapter.

Design and Design Architecture Theories

Having presented several abstract models of design and different ways of analyzing existing designs from instructional design and from other disciplines, I will now clarify the difference between instructional theories and (instructional) design theories which include (instructional) design architecture theories, and present an abstract model for a design architecture theory.

Instructional theories address ways of instructing. Instructional design theories, on the other hand, address aspects of instructional design independently of instructional theories. Instructional design theories share much in common with design theories from other disciplines and can thus fairly be referred to simply as design theories. Gibbons and Rogers state that instructional theories are applied within the larger scope of design theories (2009, p. 308). Gibbons elsewhere has described a theory of design architecture consisting of layers of a design, similar to Schön's (2009). This type of design theory or design architecture theory addresses aspects of the structure of instructional designs, also without being tied to any specific instructional theory, nor to a specific design methodology. Reigeluth and Carr-Chellman (2009) also distinguish between design theory and instructional design theory or instructional theory (pp. 7-9). They suggest that design theories focus on the process of creation of artifacts, whether abstract or material, whereas instructional theories are focused on aspects of instructional analysis. Reigeluth earlier discussed common factors in instructional design theories (1999). He posits that all instructional design theories have four common elements. They are design-oriented; they identify methods of instruction; the methods of instruction can broken into more detailed components and fourthly, the methods are probabilistic rather than deterministic, in that they increase the probability of attaining educational goals without guaranteeing goal attainment (pp. 5-7). These factors provide an insight and theoretical background into aspects of instructional design that may be found in a study of any instruction, including this study of technical instruction. They, however, are neither specific enough nor detailed enough to be used for analysis. For analytical purposes this study used and extended the idea of instructional design layers and modular design.

In contrast to design theory, design architecture theory posits that any instructional design will have identifiable architectural aspects that relate to each other in meaningful ways, without necessarily analyzing the way in which the instruction was created. A design architecture viewpoint is valuable for the current research project as it provides an independent view for identifying aspects of instruction related to higher education information technology disciplines and does not espouse or require looking for specific instructional theories nor instructional design theories.

In particular, the architectural structure proposed by Gibbons and others of design layers allows design aspects such as content and strategy to be identified and isolated in existing designs (Gibbons & Rogers, 2009). The research questions address how instructional design evolves subject to various internal and external influences and how choosing a design architecture theory approach avoids interpreting the observed phenomena through any particular instructional theory lens and without reference to any methodology, and allows the existing structural aspects of observed phenomena to be identified and interpreted.

Brand's work in design layers influenced the development of design layers in educational theory. All of these researchers, Brand in architecture, many people in computing, and Reigeluth and Gibbons in instructional design theory, argue that designs in their respective fields can be considered as layered, each layer having its own characteristics, languages, specialists and sub-layers.

Gibbons and Rogers (Gibbons & Rogers, 2009) propose a seven-layer model of design layers that can be used as a viewpoint for any instructional design. Their work was based on Gibbons' earlier work, which defines the layers as shown in Table 2.

Table 2.

Instructional Design Layers (from (Gibbons, 2000))

Layer	Function
Content	The content layer is concerned with the abstract organization of the instructional content. How will it be captured, partitioned and represented?
Strategy	Every artificial thing the designer does, all of the dimensions of all the artificial events. Deciding how to structure the course presentation is included here.
Control	The actions that the learner can perform. How does the learner speak to the system?
Message	The structure and message types that the system will use to communicate with the learner. Messages can be verbal, written, body language etc.
Representation	Motions and signs, instantiation of the messages with specific multimedia content; colors, fonts, screen arrangement, etc.
Medialogic	The logic that controls the media that carry the message to the learner.
Data management	Record keeping, progress monitoring, accumulating the record of the instruction.

These layers of instructional design, like those of Brand for buildings, are intended (among other things) to allow designers to create instructional designs whose layers can later evolve separately. It should be emphasized that, like the networking model, these layers are not a design methodology and whether or not the designer is aware of these layers when developing course curricula, these functionalities exist within the instructional design. By defining the layers and considering them at design time designers create what Brand refers to in his model as "shearing layers", i.e. separating boundaries for different parts of the course design. These shearing layers are analogous to the interfaces that separate modules in modular design.

Design layers have been further described in terms of the activities, functions and design questions that are answered on each of the seven layers (Gibbons & Helps, in preparation). This detailed analysis provides a template for exploring the questions to be addressed to IT instruction designers to indicate how changes made in courses and curricula affect the design structure.

As indicated earlier, this view of instructional architecture is independent of any instructional theory. It provides an architectural framework though which instructional designs can be viewed and analyzed. An aspect of this approach that is of particular relevance to the current research project is that conceptually, changes can be made to instruction on one or more layers with minimal impact on other layers. Since the intent is to examine changes that occur as aspects of the instruction evolve, being able to look at the changes in terms of a layered structure not only indicates how changes can be described, it promises a future opportunity for designing systems to accommodate change.

This chapter has presented a number of different theories related to instructional design and architecture, as well as related theories from other disciplines. These theories were used to analyze and describe the research data. In particular, the concepts of domains or layers of design architecture were used to analyze the data and to understand the nature of course changes within the IT discipline. Future solutions based on this approach can reasonably be expected to yield similar benefits in terms of productivity to those achieved in other fields that have used these approaches.

Chapter 3: Methodology

The methodology adopted for this research was a qualitative, naturalistic study. The study was influenced by grounded research. The research was done as a collective case study, where the case was a selection of course changes that reflected the process of course change in IT. Seven faculty members at two institutions were interviewed and the interviews were transcribed. A number of change events were identified from the interviews and the change process was analyzed. Word frequency analysis was attempted with poor success but a follow-on analysis led to recurrent themes being identified in the interviews. Domains of design emerged from the analysis and the change process was analyzed in terms of domains, or layers, of design, which were then further classified in terms of the design decisions they represented and in terms of structures, activities and functions, which is related to Structure-Behaviors-Functions (SBF) analysis. This is a design architecture representation of the results of the case study.

The justification for this methodology in terms of established methodological approaches and its specific application to the research questions are discussed in this chapter. Trustworthiness mechanisms are also discussed and included. Finally, an audit and reflection of the methodology concludes the chapter.

There were multiple possible approaches that could have been taken to answering the research questions. The principal constraints that determined the selection of research methods were the nature of the research questions and the limitations of time and researcher resources. This chapter reviews the appropriateness of several general approaches and then discuss how qualitative research and a case study approach were chosen for addressing the research questions. The role of the researcher as a factor in the study is discussed. The research for this study was performed by a single researcher. Research procedures and data gathering are outlined. The trustworthiness or “parallel criteria” related issues are discussed as they apply to this study. Finally the selected analysis approach is discussed.

Gibbons and Bunderson in “Explore, Explain, Design” (2005) suggest that research problems fall into three broad areas. Scientific research attempts to control environmental variables, perform carefully designed experiments and then use the results to explain the observed phenomena. Cause and effect characterize the design and analysis of the research approach. Design research, on the other hand, combines existing and new artifacts and ideas to create new systems, which are then evaluated against some set of agreed standards. Improvement of performance is the watchword for design. Lastly, exploratory research seeks to produce observations that can reveal relationships, suggest hypotheses and categorize observations. The intent of the research questions in this study is to explore the existing curricular evolution situation in information technology. Gibbons and Bunderson address the key aspects of the ‘Explore’ approach thus, “Within emerging domains of human knowledge the questions concern what is there and what are possible groupings and relationships among what is there.” (p. 928).

Clearly, therefore, the “explore” model of research is most appropriate for this study. This natural history approach to research does not typically produce causes and effects but leads to categorization of results within an appropriate and meaningful framework and to identification of relationships between artifacts or abstract issues identified in the research. The goal of the research was to achieve a richer understanding of possible answers to the questions and of the circumstances surrounding them. The framework that was used as the initial organizing criterion for this research study was a set of concepts around modular design, abstract design domains and design architecture theory, addressed earlier in the literature review.

Within exploratory research both quantitative and qualitative methods can be used. In choosing between a quantitative and a qualitative approach, Stake (1995) suggests that the choice is driven by intention:

> A distinction between what knowledge to shoot for fundamentally separates quantitative and qualitative inquiry. Perhaps surprisingly, the distinction is not directly related to the difference between quantitative and qualitative data, but a difference in searching for causes versus searching for happenings. Quantitative researchers have pressed for explanation and control; qualitative researchers have pressed for understanding the complex interrelationships among all that exists. (p. 35)

Since the research questions ask what is happening within the specified area, a qualitative approach seems be most appropriate for addressing these questions. The current research is not addressing clearly described design goals with desired causes and effects, which might invite quantitative methods; but the environment previously described is that of a new discipline with emerging standards and expectations. The goal was to achieve a richer understanding of relationships amongst aspects of the mechanisms at work. This study does not attempt to present a definitive view of evolving technological curricula; rather it attempts to present a thoughtful insight into the processes, effects and underlying structures that are found in such evolutions. Denzin and Lincoln describe the role of the researcher in qualitative environments as follows: "The researcher, in turn, may be seen as a *bricoleur*, as a maker of quilts, or, as in filmmaking, as a person who assembles images into montages" (Denzin & Lincoln, 2000, p. 4). They then go on to describe the multifaceted nature of the resultant representations of the research.

Frederick Erickson (as cited in Guba & Lincoln, 1989, p. 34) emphasizes the interpretive nature of qualitative research. A qualitative researcher necessarily will interpret what is observed and should also present the multiple interpretations that are inherent amongst the participants or in the system itself. Other theorists indicate the need to preserve the 'multiple realities' that can emerge in qualitative studies (Alkin, 2004; Smith, 2005; Stake, 1995; Williams, 2009). Guba and Lincoln (as cited in Stubbs, 2006) also emphasize the need for 'value-pluralistic' qualitative research, and further claim that attempts to develop 'objective' instruments involve value judgments. Denzin and Lincoln address this subject by again evoking the image of the quilt-maker, "The interpretative *bricoleur* understands that research is an interactive process shaped by his or her personal history, biography, gender, social class, race and ethnicity, and by those of the people within the setting." (Denzin & Lincoln, 2000, p. 6). Allowing and encouraging multiple viewpoints and values to emerge from this research will engender the richness necessary for a meaningful qualitative investigation.

This is a naturalistic study. Stubbs (2006) presents a description of naturalistic study methods as follows:

> Methods are considered naturalistic if the researcher does not attempt to manipulate the research setting. "The research setting is a naturally occurring event, program, community, relationship, or interaction that has no predetermined course established by and for the researcher. Rather, the point…is to understand naturally occurring phenomena in their naturally occurring stages" (Patton, 1990, p. 41). (p. 91)

Guba ((1978) cited in Patton, 2002) defined "naturalistic inquiry" as a discovery-oriented method that involves minimal manipulation by the investigator and no prior constraints on the outcome. Patton further notes that naturalistic inquiry focuses on the actual operations and impacts of a process, program, or intervention over time. (p. 104)

In the current study the research environment was not manipulated. Data were gathered from participants and artifacts in their normal work setting and analyzed to explore the research questions. The discovery-oriented nature of naturalistic inquiry fits the research questions and expected form of the results. A structured approach was used for analyzing the findings but there were no prior constraints on the outcome. For all these reasons a naturalistic, qualitative approach was used in this research study.

Resource Constraints

Resources also affect the research design. Limited resources were available for this research project. The primary limitation was that there was a single researcher. This constraint limited the possible scope of the research study. This limitation mitigated against nationwide surveys and hundreds or even tens of interviews, observations and artifact analyses. The approach that fitted both limited constraints and the exploratory and naturalistic nature of the inquiry was a case-study approach.

Case studies can use a single case to illuminate characteristics, or the boundaries of the case can be drawn around a single topic of interest. In this study the case proposed was the phenomenon of evolving curricula effects within higher education information technology programs and in order to study this either a single example or a few examples could have provided the needed information. Stake (1995) describes cases, denoted by theta (Θ) and issues denoted by iota (ι) (pp. 2, 16). A case (Θ) is an integrated system with boundaries. If we are interested in that specific case then it is an "intrinsic case study". A specific issue, (ι) may dominate a case study: a case is selected and studied to learn more about the issue. Such a case is, in Stake's terminology, an "instrumental case study". If several similar cases (e.g. a selection of instructors) are studied to examine an issue then it becomes a "collective case study" (p. 4). This most accurately describes this research project. The several subjects (instructional design changes and designers) were studied to discover information and relationships about the underlying question (the nature of curriculum change, as described in the research questions). In this situation the “case” is a specified situation or system and data is drawn from a few selected sources to provide insights into the functioning of the system. A few examples of course changes, as implemented by selected course designers or instructors, were chosen to study the phenomenon of interest. While using a single example of the phenomenon does theoretically define a case or issue, using multiple sources increases the richness of the description and improves the credibility of the study, which aids understanding of the research questions.

As stated above, the case proposed is the phenomenon of evolving curricula effects within higher education information technology. The issue of interest here is the deliberate changing of courses and curricula. The focus of the research is these changes, but more specifically the process of change surrounding the change event. The change event is defined intrinsically by the decision of a designer to effect a transition from one course state to another. The analysis was limited to intentional changes, planned and executed in a process of instructional design. It is recognized that instructors can and do introduce changes during instructional conversations and in direct response to learners’ control signals during the process of instruction. However such change events were not considered course changes for this study unless the course designer (instructor) subsequently chose to make them an on-going part of the course design. The decision to make a change, whether inspired by an instructional interaction or whether decided prior to the instruction taking place, provides an event that can be discussed, analyzed and used as source material for this study. The study considered the transition, its origins, processes, path and destination. In other words, where did it come from, what happened in the process of design and where did it arrive as an instructional design event?

The scope of the change event is a question for the research and thus was not defined as part of the methodology. The events addressed ranged from a single instructional module within a course up to a complete course re-design. While redesigning multi-course curricula are significant change events these large changes were felt to introduce too many additional variables for the scope of this study, and were accordingly not considered.

Information was drawn from interviews with designers (faculty members) and it was found that each designer interviewed had a few course change events that they were willing to describe. To keep interviews reasonable in length and scope, each interview with a course designer was limited to identify and address between one and three course changes. Since course designers were the primary source of information for course changes, they were selected as interview subjects. A larger question was which institutions or programs could provide appropriate interview subjects. Using a top-down approach I first present an argument for the institutions to consider, then criteria for selecting course designers and finally criteria for identifying course changes, the unit of interest.

Stake (1995) suggests that case studies should be selected that will "maximize what we can learn." There are only a limited number of choices of IT programs from which to select sources. The discipline is relatively new and there are only tens of IT programs in university higher education nationwide. Of these only nine are accredited at the time of writing[1]. Two of the accredited institutions, Brigham Young University (BYU) and Rochester Institute of Technology (RIT) took the lead in organizing accreditation standards and curricular models by hosting the first nationwide meeting of IT programs at a Committee for Information Technology Curriculum (CITE) meeting at BYU (Aspen Grove) in 2001 (Lunt, et al., 2002). CITE later became ACM-SIGITE and is still the organization that takes a leading role in determining and describing the

[1] See Information Technology programs at http://www.abet.org/accreditcac.asp

academic discipline of IT. BYU and RIT faculty members are still actively involved in leading roles within ACM-SIGITE. Because of their deep involvement in developing IT curricula and programs, RIT and BYU were selected as source organizations for data gathering for this study.

There were other reasons for selecting these two sources. Typical cases are often beneficial as are case studies that are accessible and "hospitable to our inquiry" (Stake, 1995, p. 4). The information technology programs at BYU and RIT qualify in these terms. BYU's IT program is very accessible since the researcher is embedded in the BYU IT program. RIT was not physically as easily accessible to the researcher as BYU, but he is personally acquainted with some of the RIT faculty members and a visit was relatively easy to arrange. Sufficient data were gathered in a visit to RIT lasting a couple of days. The visits, combined with follow-up communications by email and phone, provided sufficient data for this study.

Having selected appropriate institutions, selecting designers to participate was relatively easy. The criteria were that faculty must have implemented one or more identifiable changes in their information technology courses, be willing to discuss these in an interview and be willing to provide necessary supporting materials. Taking advantage of the researcher's personal relationship with the selected programs it was possible and reasonable to send a request to all the faculty at these two institutions asking for cooperation. As expected, a suitable number of interview subjects were obtained with this simple procedure. The selected subjects had varied experience and diverse backgrounds. Identifying specific course changes took place during the interviews.

In summary, the process of course change events was studied. Interviewing several IT professors at each of the two identified institutions and discussing a few change events with each of them provided a variegated sample of course change events. There was sufficient variety among both professors and events to yield a meaningfully diverse result.

Role of the Researcher

An important aspect of qualitative research is that it recognizes the role of the researcher within the research environment. Classical scientific research attempts to define and use a completely objective viewpoint, but qualitative research recognizes that objectivity, particularly in social sciences, is extremely difficult and probably impossible. Lincoln and Guba (2000), amongst other researchers, discuss the issue of control and objectivity in research, and suggest there are distinct benefits in the interaction of the researcher with the research subjects. They suggest that not only is objectivity in social research unattainable but that involvement leads to better involvement of the community being studied, generation of possible solutions and better interpretations of the research questions. (pp. 175, 176)

In this case the researcher functions within the following roles that impacted the research study. Firstly, I was the sole researcher for this study with a background as a graduate student in instructional technology. Secondly, I am an instructor and associate professor within the BYU Information Technology program, one of the programs selected for this study. I have twenty years experience in designing and teaching courses in Electronic Engineering Technology and Information Technology. Thirdly, I am an accreditation evaluator and team chair for ABET. As an accreditation evaluator I have visited multiple campuses and have reviewed programs in Engineering Technology, Information Technology and Computer Science for accreditation purposes. This experience provided thorough access to faculty, detailed curricula, facilities and other information about programs in these fields.

The researcher's experience and viewpoint were reflected in the research study. There were positive benefits to this viewpoint, which included an understanding of the domain-specific subject area, leading to a quicker understanding of issues revealed by the study. The researcher also has a working relationship with faculty members in the study field, which facilitated what Fontana and Frey call, "accessing the setting" (2000, p. 654), by which they mean the ability of the researcher to be accepted by those within the community being studied. This is not a problem because the researcher is a member of that community.

As a researcher my background as an insider in the field may have also negatively impacted the study. For example I may have assumptions and expectations of how things *should* work, which would not bias an external researcher. Procedures to protect against the negative issues will be discussed in the trustworthiness section below.

Data Gathering

"Much of what we cannot observe for ourselves has been or is being observed by others," says Stake (1995, p. 64). A primary source of information for this study was interviews with those who design and change IT courses. The process of change was not observed but those who have performed these changes multiple times were interviewed. These were the instructors, sometimes in collaboration with teaching assistants or campus instructional development departments or experts, who made the decisions, designed the change and implemented the courses. In all cases the instructor was at the center of the curriculum change, and was the protagonist.

Interviews can be structured or unstructured (Fontana & Frey, 2000). Structured interviews are strongly controlled, in an attempt not to influence the subject by injecting the interviewer's viewpoint – a goal at odds with the expectations of qualitative research. Unstructured interviews, on the other hand, are open-ended, allowing the subject and interviewer interactions to determine the results. Stake (1995) also states that case studies obtain the descriptions and interpretations of others and that, “The interview is the main road to multiple realities.” (p. 64) He agrees with Fontana and Frey in stating that structured interviewing with formal questions and rigid interview protocol is not appropriate for qualitative studies. In qualitative studies each person interviewed is expected to have had unique experiences and individual viewpoints. The approach described by Stake (1995, p. 65) was used for this study; specifically the interview was guided by a short list of issue-oriented questions. Each interviewee was encouraged to develop individual responses to these questions.

Interview structure. In line with the research questions, the short list of questions used for the interview addressed *inter alia*, courses in IT that the interviewee was involved in designing, the background and history of the course design, how courses are developed in the department and institution, who creates the courses and who changes them. The discussion proceeded to address what courses were changed, the granularity of the changes (complete courses or minor updates), and causes and consequences of the changes. This was followed by a discussion of the effects of the changes, specifically what other changes were caused by the initial changes, what impact this had on the educational structure and design architecture, and to what extent the instructional architecture was a consideration in either the original design or in the changes. The interview also addressed how the process was managed, whether multiple people were involved and what roles these additional people played, how the process was coordinated and what the governing principles of the change process are.

Negotiated interview results. Fontana and Frey (2000, pp. 661-663) indicate that researchers need to, “‘get to know’ respondents beneath their rational facades” and that they should be willing to engage with them. The necessary complement to this approach to research is that the role of the role of the researcher in the research process must be explicitly identified and discussed. In this case this researcher has significant personal interest in the research question. In addition to my personal role as principal investigator in this instructional architectural research I have personally dealt with many of the difficulties cited as inspiring this research. Other researcher roles were discussed previously.

The interview questions were tested in pilot form, first by the researcher and then with a volunteer test subject. This pilot interview showed that an open-question format did elicit responses from the test subject relevant to topics of interest to this study in his own terms and using his own vocabulary. The test subject talked about course changes, what motivated them, how he managed them and so forth. These results were superficially evaluated and used primarily to validate the procedure. This indicated that an open-question format would produce usable results. The interview protocol was modified slightly and then used for the formal interviews recorded in this study. Interviews were audio-recorded. Originally the interviews were summarized, rather than transcribed word-for-word. Stake says, “Getting the exact words of the respondent is usually not very important, it is what they mean that is important.” (p. 66). This decision was changed later, as discussed in the procedures audit section of this report. In this study it was not immediately apparent which words were the ones that conveyed the true sense of what the respondents meant or how that affected the underlying design, so the audio recordings were later completely transcribed. Summaries of the interviews were submitted back to the respondents for their comments and corrections as part of the validation process. This too is discussed in the audit section of the report.

Stake (1995, pp. 51-57) also provides detailed guidelines for conducting case studies, including schedules, stages and checklists. These guidelines provide further information for data gathering. A data gathering plan based on these guidelines is contained in Appendix A.

Artifacts. Arising from the interviews, artifacts which impacted the course design and change process were identified, such as course outlines, student assignment sheets, institutional course design guidelines, books, journal articles, supplementary websites and other items. Copies of these were requested where appropriate. These were augmented with photographs of working environments but these photographs did not provide much new information about the course change process and were thus not included in the results section. These artifacts were considered along with the interview results to generate responses to the research questions.

IRB approval. Because this research involved interviewing human subjects IRB approval was sought and obtained.

Trustworthiness

The traditional paradigm of research quality requires rigor in carefully and objectively isolating effects and linking them to causes. Guba and Lincoln (1989) suggest the conventional scientific criteria for rigor are: internal validity, external validity, reliability, and objectivity. In a constructivist inquiry Guba and Lincoln identify "parallel or foundational criteria" for ensuring the trustworthiness of the inquiry. Their four parallel criteria are: credibility, transferability, dependability and confirmability (pp. 233-243). Each of these criteria have been considered and addressed in this research project.

Credibility. Several techniques are identified for improving credibility (Guba & Lincoln, 1989; Janesick, 2000; Williams, 2009). Some of these were used in the research project. The techniques that are proposed in various forms include (a) member checks, (b) emic perspectives, (c) prolonged engagement, (d) persistent observation, (e) progressive subjectivity checks, (f) triangulation, and (g) use of the Program Evaluation Standards. These will each be discussed as they apply to this project.

A powerful technique that is relevant to the current inquiry is that of member checks. The researcher tests hypotheses, data and interpretations with the participants who provided the data originally. This was used in the research project by exchanging and discussing portions of the inquiry results with the interview participants following the interviews. Their comments and assessments of the summaries were invited and the transcripts were annotated and refined based on their feedback. In a note of caution Pickard and Dixon (2004) warn that member checks should not 'marginalize the researchers' voice.' Thus member checks strengthen the credibility of the inquiry, but do not dominate it.

Including participants' words in the results can broaden the perspective of a qualitative inquiry i.e. an *emic* rather than an *etic* perspective (Vidich & Lyman, 2000; Williams, 2009). Data gathering for this study used an open interview format. Although a short list of questions framed the interviews, participants were invited and encouraged to present their viewpoint in their own voice. This emic perspective was reinforced through the member check technique. Asking participants to validate the outcomes of the interviews they participated in ensured that they felt their own ideas were being expressed. Selections from the participants' own words and artifacts are included in the written study results.

Prolonged engagement is a technique that requires a substantial involvement at the site of inquiry and leads to less misunderstanding of observed phenomena. The researcher has been immersed in the department being researched at BYU for years and has visited the department at RIT several times and has interacted with faculty members multiple times over several years. Thus many of the benefits of prolonged engagement are already available for this study.

Persistent observation adds depth to the scope of the inquiry by focusing on relevant items in the research data in detail. For this project it was achieved by the thoroughness of the analysis of the interviews and artifacts.

Progressive subjectivity checks reveal the researcher's changing perspective as the inquiry progresses and are an indicator of the lack of bias in the inquirer. These were pursued by writing comments on the progress and status of the research and findings throughout the inquiry. These comments formed the basis of the audit section of the research procedures during the analysis phase of the inquiry.

Another standard technique for increasing credibility is that of triangulation. Triangulation implies that several research vectors are surrounding a single truth, whose exact location is approximated by the triangle formed by the intersection of the vectors. Janesick (2000, p. 391) proposes crystallization as an alternative to triangulation. Crystallization also suggest multiple viewpoints but instead of implying a single truth that is being identified it uses the metaphor of a crystal, whose many facets illuminate the subject in alternative ways. It thus moves even further from the objectivist paradigm of a truth to be found towards the constructivist view of a multi-faceted issue to be understood constructively. In this research several different types or sources of information were gathered to achieve the benefits of triangulation, or crystallization, that address the research questions. These are (a) interviews from multiple participants (seven participants were interviewed), (b) course outlines, (c) sample lesson plans, (d) descriptions of teaching and learning environments, and (e) other artifacts from discussions with participants.

The Program Evaluation Standards published by the Joint Committee on Evaluation standards are also relevant as a practical guideline for ensuring and improving credibility. These standards are intended for evaluations rather than a case study, such as this one, but their practical approach offers good general guidance. The Utility, Feasibility, Propriety and Accuracy standards were used as a checklist for this research to the extent that they were applicable (Joint Committee on Standards for Educational Evaluation. & Sanders, 1994). Comments on these guidelines are included in the audit section.

By employing the above techniques the credibility of the research is established.

Transferability. Transfer of results can only be done by others, but transferability can be enabled by providing thick descriptions. Pickard and Dixon (2004) describe transferability in case studies as, "rich, descriptive narratives at a micro level to provide detailed descriptions which will allow readers to make sufficient contextual judgments to transfer outcomes, themes and understanding emerging from the case studies to alternative settings." Transferability can be further enabled by providing readers with a view of the potential benefits of the study. This may motivate them to seek similar benefits in their own situations, thus encouraging them to transfer the results of the study.

Dependability. An audit trail of the research was created by (a) records of all data gathering, (b) summaries of interviews, which were also used for the member checks, (c) status summaries, and (d) records of changes in methodology that occurred as the research proceeded.

Confirmability. Confirmability or quality is ensured by two mechanisms. Firstly the research is grounded in current, recognized research practice, as evidenced by the literature review. Secondly others, primarily the graduate committee, reviewed the complete project.

Analysis of the Data

Christopher Alexander (1979) describes an approach and a viewpoint to understanding a structure which is applicable in concept to understanding the results of this research.

> Suppose I want to understand the "structure" of something. Just exactly what does this mean? It means, of course, that I want to make a simple picture of it, which lets me grasp

> it as a whole. And it means, too, that as far as possible, I want to paint this simple picture out of as few elements as possible. The fewer elements there are, the richer the relationships between them, and the more the picture lies in the "structure" of these relationships. And finally, of course, I want to paint a picture which allows me to understand the patterns of events which keep on happening in the whole structure I seek. In other words, I hope to find a picture, or a structure, which will in some rather obvious and simple sense, account for the outward properties, for the pattern of events of the thing which I am studying. (p. 81)

In an analogous manner the analysis of the data should provide a simple picture revealing the structure of how courses change over time and under the effect of various influences. Ideally the analysis should combine a few elements but include rich relationships. This is achieved by using theoretical structures that do not constrain the expected results to specific instructional theories. The concepts of design frameworks, domains, and layers enable this viewpoint. The design domains paradigm provides a structure for '*grouping the findings*' . Assertions will be generated from these findings and '*patterns of generalization'* will be sought (Alexander, 1979). Information derived from the interviews reflects back to the architectural analysis, and extensions or modifications to the architectural structure can be inferred.

Edelson (2002) discusses using design as a research tool and indicates that the "design research paradigm treats design as a strategy for developing and refining theories." He characterizes the theories that can be developed through design research as 'domain theories', 'design frameworks' and 'design methodologies'. He describes domain theories as follows:

> A domain theory is the generalization of some portion of a problem analysis. Thus, a domain theory might be about learners and how they learn, teachers and how they teach,

> or learning environments and how they influence teaching and learning. For example, a curriculum designer might develop a domain theory about the needs of learners or the challenges of implementing a type of learning activity in certain settings. Even though a domain theory in design research is developed through a design process, it is a theory about the world, not a theory about design per se. As such, it is descriptive, not prescriptive. Design research can contribute to two types of domain theories: context theories and outcomes theories. A context theory characterizes the challenges and opportunities presented by a class of design contexts. (p. 113)

In this case the study is analyzing existing designs, not specifically engaging in design to develop theories; however, the results from the study generally fit the description of a domain theory as defined. Designers design course changes and curricula to meet their changing needs. The overall design architecture paradigm and the designers' usage guidelines are revealed by this research. More specifically, the instructional design architecture theory and analysis enable relatively detailed understanding of the possible outcomes without constraining the results by assuming specific instructional theories or instructional design methodologies.

As the interviews were completed during the data gathering phase of the project, the design questions in this paper provided a sub-text to guide understanding of the phenomena observed. Post-interview analysis interpreted the results to answer the research questions. Design domains or layers were allowed to emerge from the data to create a structure describing the design architecture. The interview protocol is included in Appendix B.

Analysis protocol. Qualitative research analysis must generate dependable results. Patton (2002), echoing other theorists, describes dependability in qualitative inquiry as “a systematic process, systematically followed” (p. 546). Systematic processes have been developed and described by several theorists in qualitative research. There is, however, a tension in qualitative analysis between using well-recognized and proven analysis practices and adapting the research methodology to suit the needs of the qualitative environment and following discoveries within the data. Tesch (1990) describes a number of recognized forms of qualitative studies and then states, "… individual scholars have conducted inventive qualitative studies without labeling their method. Each one made up her/his own way of analyzing data." She further states that, "No one has 'codified' the procedures for qualitative analysis and it is not likely that anyone ever will. Qualitative researchers are quite adamant in their rejection of standardization." (p. 4)

On the other hand, qualitative inquiry must be pursued within a body of recognized practice. The alternative to using recognized practice as a starting point is that researchers must not only develop their own theories and practices, but also justify them fully to validate their trustworthiness and dependability.

In this research project this tension was addressed by using accepted methods of qualitative inquiry as a starting point and then adapting them to suit the needs of the current inquiry and research goals. The deviations from the standard approaches are identified so that the reader may evaluate their validity for this application and decide how this may affect the transferability and dependability of the results.

The methodological analytical approaches that resonate most strongly with the research questions are those of grounded theory and schema analysis.

Glaser and Strauss developed a theory and a methodology of discovering theories in sociology by analyzing data, which they called "grounded theory" (Glaser & Strauss, 1967) Ryan and Bernard (2000) say that grounded theorists desire to "understand people's experiences in as rigorous and detailed a manner as possible" and to "identify categories and concepts that emerge from text and link these concepts into substantive and formal theories" (p. 782). Methodologically, "Grounded theory is an iterative process by which the analyst becomes more and more grounded in the data and develops increasingly richer concepts and models of how the phenomenon being studied really works." (p.783).

Schema analysis is slightly different in intent although similar in method to grounded theory. Ryan and Bernard (2000) state that, "It is based on the idea that people must use cognitive simplifications to help make sense of the complex information to which they are constantly exposed." (p. 783). Ryan and Bernard in a later paper (2003) list a series of techniques for analyzing qualitative data and indicate that researchers need to select among these techniques, possibly using more than one of them as the data is analyzed.

The difference between what these approaches do and what I used is that in this research there is a concept that is being used as an analytical framework. Questions are not being asked to see what theories of curriculum change might emerge, but the questions are directed towards seeing what effect the changes might have on the underlying design architecture and using the set of concepts of design domains, layers and design decisions, as discussed in the literature review chapter, as a working paradigm to identify the architectural effects. Thus, as in grounded research, theories of the change process were expected to emerge from the research but the concepts of design architecture, to some extent, overlaid and constrained those theories. This relates to the schema analysis approach in that the idea of domains or layers was used as a pre-existing cognitive interpretation of the changes. However, the interview questions were very open-ended and the list of interpretative questions was specifically intended to be very open-ended. Thus, as expected, the results were not overly constrained by the initial choice of analysis paradigm.

The analysis methodology used here was similar to that for a grounded theory or schema analysis inquiry, adapted from Ryan and Bernard (2000, pp. 782-785; 2003). A protocol was developed for analyzing the data. The protocol included transcribing the interviews, identifying course changes and then reading the transcriptions multiple times looking for common words, phrases or ideas. Themes and architectural structure began to emerge and these were refined and improved until coherent and consistent themes and structures were articulated. Participants' words were used to express the themes and structures. A more detailed description of the data analysis protocol is in Appendix C.

Presenting the results. When theories or models of change are created they can be expressed in several alternate forms. These forms were necessarily adapted to the findings. The outline below presents the general form of the presentation of the findings.

The events of change can be effectively presented through descriptions surrounding the change processes. The descriptions refer to the theories that emerged from the analysis and attempt to paint a picture or tell a story that conveys the process and thinking of the course designers as they evolve their courses. Impacts of teams, external influences, costs and project management are addressed within this description, as well as the issues related to the original research questions and architectural changes that emerged. The description also includes quotations and other original source material that illustrate the perspective of the course designers.

In addition to the above description of the changes, the design architectural aspects are discussed. The extent to which the design domains or layers theory provided a coherent framework and vehicle for discussion of instructional design is addressed. Finally, additional sections summarize answers to the research questions. These sections refer back to the descriptive material but are more focused on the questions than on the changes.

Audit of and Reflections upon Research Procedures

In order to ensure the trustworthiness of this research a number of aspects of the 'parallel criteria' were used, as discussed previously. In particular, several specific measures were applied to improve the credibility, as defined by the parallel standards. The techniques that were proposed and used in various forms include (a) member checks, (b) emic perspectives, (c) prolonged engagement, (d) persistent observation, (e) progressive subjectivity checks, (f) triangulation, and (g) use of the Program Evaluation Standards.

Member checks were created in the form of summaries of each interview, which were sent to the participants. They were invited to correct, amplify or confirm the information summarized there. In general they confirmed the summaries; in a few cases they clarified some issues or provided small amounts of additional material. These were all incorporated into the revised summaries. Copies of the revised summaries are contained in Appendix E

Emic perspectives have been used by quoting the participants and reflecting their own viewpoints wherever possible. Prolonged engagement and persistent observation were both employed in the interview and analysis phases. Progressive subjectivity tests were performed by checking with external observers periodically. Triangulation was made possible by the multiple participants responding to the same issues and the multiple quotations or artifacts that were used to explore issues. Finally the Program Evaluation Standards were applied as a guideline to check the procedure.

The Program Evaluation Standards (Joint Committee on Standards for Educational Evaluation. & Sanders, 1994) also provided a measure of credibility for this study. Each of the four general categories of Utility Standards, Feasibility Standards, Propriety Standards and

Accuracy Standards were considered.

With reference to the Utility Standards, stakeholders, evaluator credibility, information scope, values and impact were all considered in the discussion of research methods. Most of these issues were further considered in the IRB application process. With reference to the Feasibility Standards, practicality, viability and cost effectiveness were addressed in the research proposal. With reference to the Propriety Standards, service orientation is addressed in the introduction to this study, formal agreements, rights of human subjects, human interactions and fiscal responsibility were addressed in the research proposal and in the application for IRB research on humans; disclosure and conflict of interest are discussed in the methodology section of this report. With reference to Accuracy Standards, each of the subcategories was considered and this report was reviewed to ensure that they were addressed. In summary, the Program Evaluation Standards as a whole have been considered and satisfied at various stages of this research study.

Transferability of results can only be done by others; however, transferability is supported by thick descriptions, which have been used throughout the results and analysis sections of this study. Dependability is achieved though this audit report and the various reviews of this study, and confirmability is achieved through the validity of the research methods and the approval process for the proposal and the IRB application.

Change from summarized interviews to fully transcribed interviews. Some aspects of the research procedure were changed during the study. Originally I proposed to create a detailed summary of each interview rather than a word-for-word transcription. I originally believed that a detailed summary would serve the same purpose as a complete transcription and that concepts and themes could be derived from these summaries. In fact, as part of the verification process summaries were prepared and were sent to the participants for their feedback, but in the process of preparing these summaries it became apparent that the emic perspective was being lost or at least distorted. Nuances of expression were being summarized and compressed from the viewpoint of the researcher, rather than that of the participant.

The procedure was accordingly changed. All of the interviews were transcribed word for word and then analyzed using word analysis software specifically designed for qualitative inquiry (HyperRESEARCH). It is worth commenting that this was an extremely time-consuming procedure. The ratio of spoken interview time to written transcription is about ten or fifteen to one for an untrained transcriber, i.e. it takes ten to fifteen hours to transcribe each hour of recorded interview. Trained professional transcribers can achieve ratios closer to two to one. The effort of transcribing becomes a limitation in doing further research.

Once the interviews were fully transcribed and were being analyzed, a secondary weakness emerged. Sometimes concepts were conveyed indirectly by verbal intonation or by body language. For example, in an interview Adam says, "It's like every other class that I … that doesn't vary very much." The researcher in this case was able to fill in the missing word and information from context and from memory of the conversation to give, "It's like every other class that I [develop]. That [meaning the teaching style] doesn't vary very much." However this came from the participant gesturing towards his development process and documents and is not recorded in the transcription. In this particular case the consequences of not having a record of the body language are not very serious. The word, "develop" could be replaced by "teach" or "evolve" without significantly changing the conclusions drawn from the statement, and surrounding material in the transcription supports this interpretation. In the more general case misunderstandings can easily arise, particularly if the interviewer and the analyzer are not the same person. A related situation occurred with Susan who used multiple instances of non-English sound effects to communicate. For example she would growl (“Grrrr”) to signify frustration and say a sound like “pah-pah-pah” to indicate a rapid sequence of events, and she would say “blah-blah-blah” to indicate a filling in of verbal detail not explicitly described. Similarly Tom says, “they're formatted; da, da, da, da, da,” with his hands indicating stepping down the elements in a list. This non-English verbal communication was effective in a face-to face situation but difficult to transcribe and difficult to interpret to convey the emotional content of the communication. Body language wasn’t transcribed at all, merely remembered. These incidents reflect two conflicting problems with research of this type. Firstly, the examples suggest that it may be desirable to record interviews with as much richness as possible. Perhaps these interviews could have been enhanced with a video of the proceedings; probably two cameras would have been necessary, one to focus on the participant and one on the changing material on their computer screen or desk. Competing with this natural desire to more fully explore the phenomena being studied is the problem of having to analyze this hugely enhanced data stream. How would video be “transcribed?” What notation would be used for emotional emphasis? How much additional effort would be required to analyze this deep data stream? We now have cheap, reliable and high-fidelity video recording equipment to enable the rich recording of interviews but there is a dearth of automated tools for processing and analyzing the results. No immediate solution to this problem suggests itself, merely the hope that in future better analysis tools will emerge to simplify the process and thus enable richer data streams to be captured and analyzed.

In this latter regard some attempts were made to automate the transcription of the interviews for this research with computerized voice recognition software. The results of this attempt were not successful. An Internet search did not identify any voice recognition packages that were well accepted in this field. Several software packages that claimed to be able to do voice recognition were tried and all of them proved difficult to set up and use and the results were uniformly disappointing. Some further investigation into the research literature revealed that currently the best automated transcription systems, working under good conditions, can recognize about 80% of words (Fiscus, Ajot, & Garofolo, 2007; Pallett 2003; Wikipedia, 2010). Under normal office environments, with multiple speakers and an untrained recognition system the success rate is generally worse. This effectively requires human checking and re-transcribing substantial portions of the interview; in other words it is not very satisfactory. Automated processing of richer data streams is worse. Consequently, human transcription was used for this research.

Keyword frequency vs. concept frequency. Looking for themes by keyword frequency, as recommended by many qualitative researchers (Ryan & Bernard, 2000, pp. 79, 80; Tesch, 1990) was not very successful for this study. Tables were drawn up, classified by general concept and keywords related to those concepts were developed and then the text for each participant was searched for these keywords. In “What and How do Designers Design” Gibbons (2000) suggests a number of common layer design constructs, defined by key words, which served as a useful starting point for keywords that relate to different aspects of instructional design architecture. Other keywords arose from repeated readings of the text. These words too were searched for in the transcripts of each participant. All occurrences of keywords were counted. This exercise certainly resulted in better understanding and the multiple readings of the text provided considerable familiarity with the data relative to the research topics but the keyword frequency did not itself yield many useful results.

Two different effects led to the failure of the keyword frequency approach. Firstly there was verbal ambiguity. For example the word “fact” suggests the concept of a learnable artifact, something specific that the instructor or designer wishes to convey to the learner, a single factual idea. However, in the analysis of the transcripts the phrase “in fact,” as used for verbal emphasis, occurred multiple times, more so than any use of the word “fact” as applied to teachable artifacts. These non-relevant uses of key terms were eliminated to leave only the remaining (assumed relevant) uses of the word or phrase. This multiple-meaning problem occurred for several of the words selected as keywords, and sometimes eliminating one or more occurrences of the word from the count was a judgment call, which means that the keyword count was influenced by the viewpoint of the researcher. Having eliminated the alternate and ambiguous meanings of keywords, the remaining frequency counts were not very high. Among tens of keywords searched through for each participant the highest counts were for 13 and 12 repetitions of the words “show” and “concept” respectively. Most word frequencies were in the range of two to four repetitions each. This rendered the results somewhat unreliable for identifying important themes.

A second effect also came into play here, arising from the embedded researcher and the qualitative nature of the enquiry. If the participant mentioned an idea that seemed likely to be relevant to the enquiry the interviewer, being aware of the potential value of this concept, sometimes asked them to expand on the idea in more detail. Thus there would be multiple uses of a term or phrase, but these did not indicate a natural importance of the concept as much as it indicated a researcher-led emphasis of the topic during the interview.

Consequently, simple word frequency approaches were largely abandoned. An alternative approach of *recurring themes* was used. This is in line with, but not identical to the approach suggested by Corbin and Strauss (2007). They refer to open coding and categories or themes. Recurring themes is a similar concept but here themes are selected and highlighted if they occur in multiple interviews or multiple contexts. The earlier work from the word frequency exercises was used to suggest concepts or themes and then the transcripts were searched to find parallel or contradictory examples of these themes in other places. This led to the collection of recurring themes reported in the results section. Wherever they were available quotations from as many participants as possible were sought to illustrate the identified theme.

Finally, to improve the confirmability and credibility of the observations a secondary standard practice was added to the analysis and identification of recurring themes. After a number of related quotations were found, a theme would be identified and named. Then each of the quotations for that theme was re-read in context in the transcription to ensure that it still supported the theme to which it was attached.

Chapter 4: Results and Analysis

This results section discusses the findings of this research. Firstly, some general results describe the institutions, the participants, and the general findings to provide a context for the remainder of the discussion. Secondly, themes emerged from the data that provide a background for addressing the research questions. These are presented and discussed. Thirdly, the effects of the changes in curriculum on the structure of the instructional design are considered. Finally, having analyzed the data, the research questions are then reviewed and the findings relative to each question are summarized.

Data Gathering and Context

Seven faculty members were interviewed using the interview protocol. These faculty members were all professors of various ranks (from assistant professor to full professor), at two different institutions and in three different departments. The three departments included two that identified themselves as Information Technology; the third was a computer game development program, which had recently broken away from the IT department to form a separate department. This latter department is also considered an IT department, as it has the same core curriculum and philosophical approach as other IT departments. All the faculty from this department had come directly from the parent IT department.

The seven participants were interviewed face to face and the interviews were audio-recorded. The interviews were all fully transcribed. The professors' names were removed and replaced with alias names to preserve anonymity. Throughout this report professors will be referred to by their alias names. The seven alias names are Adam, David, Lisa, Tom, Geoffrey, Susan and Jack.

The focus of discussion in all of these interviews was changes that occur in the curriculum of IT courses. Discussion ranged around motivations, causes, process, results and implications of these changes. The interviews were analyzed to identify both significant events and themes of change. More than twenty change events in courses were identified from the interviews. They are listed in Appendix D. The changes ranged from minor adaptations to suit changing teaching strategy or minor updates of technology, up to complete course redesign projects. The changes in this list will be discussed in more detail in subsequent sections. The process, motivations and effects of these changes will be explored.

General Results

Some expected results emerged from the interviews and from the change events, such as the fact that all faculty interviewed change their curricula frequently. Some professors change courses very frequently; one professor indicated that he had redesigned or significantly modified seventeen courses within the past five years. The fact that he offered this statistic spontaneously indicates that he is very aware of changing courses. Another professor indicated that he had changed or redesigned every course he was responsible for—also in a relatively short period of time. It should be emphasized that none of these professors were full-time instructional designers, but were normal university IT faculty with teaching and research expectations.

Professors did support the concept that constantly updating their curricula placed an extra burden on them. As participant Geoffrey expressed it,

> [Our institution] is moving towards doing more formal research and expecting more publication and grantsmanship *(sic)* from the faculty, and so it's been made very clear to people that the reward system is going to lean heavily towards publication and grantsmanship and curriculum development is not always related and not always the best path towards that.

Another instructor said her method of keeping current with new technology is only successful, "because I work 75 hours a week." These insights emphasize the reality of the problem of constantly updating the curricula. Aspects of and inter-relationships relating to the processes of change were discovered by finding commonality in the data for the participants.

Recurrent Themes

Initially an attempt was made to search the data for keywords or phrases and then use the frequency of occurrence of these keywords to identify common or important concepts; however that approach did not yield satisfactory results. The methodology was modified to identify common concepts or recurring themes in the data. These changes in analysis approach are discussed in more detail previously in the section on auditing and reflecting on the methodology.

Several common themes were found. All or nearly all of the participants referred to a common set of ideas or concepts in the process of designing courses. Although they all reflected on the same themes they had different approaches to or impressions of them. A better understanding of the process came from looking at the different viewpoints of these common themes. Accordingly, quotations reflecting each of the recurring themes are used to illustrate, understand and contrast these themes, and how they impacted the change process. Some of these themes were unexpected and revealed new insights into the process; others were anticipated as they followed directly from questions in the interview protocol and were deliberately brought up in the interviews; however, even with these expected themes the different responses of the participants provided new understanding of the research questions.

Constant Change

The first recurring theme, which was expected, is that all those interviewed acknowledged constant technical change as a factor in Information Technology and in their instructional design. More revealing were the differences in attitudes towards the on-going process of technically driven change. Initially it was assumed that professors and students would be enthusiastic about new technologies and would be anxious to explore them. The actual attitudes range from considering change as enjoyable and a privilege to considering it an on-going duty and even considering it to be an unwanted driving force. These attitudes are indicated by the following quotations from the interviews (see Table 3).

Not only is there a range of attitudes but there are some mixed attitudes. Geoffrey, David and Jack are clearly enthused about constant change, Tom and Adam in these quotes (and elsewhere in their interviews) indicate that it is just part of the job, while Lisa comments that she is grateful that she no longer teaches a topic (computer coding) that requires her to update her course frequently, but on the other hand is glad to explore new media and make it available to her students. Susan apparently feels that change is inevitable, out of her control and somewhat overwhelming. Clearly, all the professors perceive that the change is necessary, and also there is a spectrum of attitudes towards the on-going curricular change.

Preferred Teaching Style Adopted and Maintained

A common theme that was not anticipated, although perhaps it should have been, is that faculty members frequently change their courses to suit their preferred teaching style. At least four of the seven professors interviewed at both institutions redesigned their courses to follow a form of the studio-teaching model. Lisa has even created a verb, to “Lisafy”, which is a corruption of her name, to describe how she has modified every course she teaches to include many graphic elements and fewer text pages, which she prefers (‘Lisa’ and ‘Lisafy’ are aliases of her real name and the verb she created). She also adopted a teaching philosophy that she describes as “Call and Response” to reflect her musical interests, but it could more prosaically be

Table 3

Attitudes to on-going technical change among participants

Alias	Quotations from the interviews
Geoffrey	I feel like I'm in a discipline that changes and has changed so much in the time that I've known it, that it's just a way of life. … And so, I feel kind of fortunate to be part of all that. It's very exciting.
David	I have a web development course that is a constant state of flux. [It's] a labor of love for me, I put in an inordinate amount of work on it, constantly. And partially because it's a very core part of IT, I feel a compulsion to make this an outstanding course.
Jack	[Curricular change] … to me [is] one of the most exciting things that we do here. Having the chance to shape what our students experience, for me, it's a compelling part of this job. Something I actually enjoy doing.
Tom	IT changes so fast and updates so much that, … you've got to stay up on top of things IT changes so much, that I think that was another thing unique to our design [ed. our discipline], and I think that is a key fundamental change that you don't get in [other disciplines]
Adam	I update my lecture notes every time I read an article that is relevant. I usually read during lunch and if I see something that is relevant I immediately update my lecture notes and change the date on the notes.
Lisa	Stuff under the hood changes a lot. Thankfully, I've been moving away from teaching any kind of coding, … Although I'm still having to relearn things. …. I think it's great that my job requires I consume media.
Susan	I don't think we manage it. I think it manages us. I think we are continually running ahead of the snowball.

described in terms of question and discussion. Although it should be mentioned that professors also incorporated new domain-specific content changes at the same time as the new teaching model – and attributed the cause for the change to the need to update the technology, it appears that they would have made the change even without technical updates being incorporated. They felt it suited their personality better and believed that it improved learning for the students. One finding of this study is therefore that professors will change a course to suit their preferred teaching style, and then generally maintain that teaching style (and some part of the course structure) for some years, even though the technical content in the course will be changed more frequently. Professors indicated their preferred teaching approach in the interviews, as shown in the quotations in Table 4.

It is interesting and relevant to note that student learning problems motivated changes in at least two of the cited cases. (David and Jack). Tom also indicates that his adoption of his preferred instructional approach was based on his understanding of student needs. Adam has a teaching style that he bases on his educational background while Lisa describes how she modified the classes she taught to suit her preferred style of teaching.

Lisa works in a sub-discipline of the program focused on graphic arts. Like other professors interviewed she regularly updates her course materials. However, she does not spend her time updating labs to incorporate the newest technology. She rather seeks out new graphic images or themes to inspire her students to create new types of projects. She expects and relies on the lab management staff to update the software in the labs to match current industry standards and she remains reasonably current with these standards, but focuses her attention and the attention of her students on the end-product of their projects and regards newer technology as a tool to get there. Unsurprisingly, students will sometimes wish to use a newer technology to

Table 4

Preferred Teaching Approach of Professors.

Alias	Relevant quote(s)
Geoffrey	I admit that course for me has been really kind of interesting, in terms of the style of how I teach, some of the ideas that are in there are certainly active learning, so I teach almost always in the lab. …. I can't say that's the model of every class but it's the characteristic
David	The course as a whole has had, historically had a third or more of the students have failed it every semester, because it is a very intensive course. And we've been trying everything we can to help. We love the course. The students who make it through love the course, but we would like to increase our success rate without diluting the course. So we've been trying lots of things, one of them is the studio teaching—
Jack	Change is driven by perceived student needs; "areas where students struggled." I've changed probably about seventeen courses here since day one. [This was in] five years. I have been involved with different rewrites, creating new courses from scratch, to modifying and evolving courses. [Jack proceeds to discuss the "Active Learning" model he has adopted for his classes.]
Tom	One of the courses I took [as a graduate student] was Training System Design. It focused around the idea of knowledge, skills and abilities. I use that a lot, because IT, when they walk out they need to … be able to do stuff and solve problems, so you've got to have skills, and you've got to have knowledge, and then abilities, it's essentially the combination of the two *(sic)* to solve problems.
Adam	[My teaching approach is] like every other class that I [develop]. That doesn't vary very much. Because I have a background in education I try to use multiple approaches so that it meets multiple learning styles.
Lisa	The structure's generally "call and response". I will go and lecture and demonstrate, and then they stay in the room and work. I require that they stay and work with their peers, because it really is a studio class, they get to feed off each other. [Combined with this teaching approach Lisa describes how she 'Lisafied' all her teaching materials. i.e. converted them all to her own personal style, which is], "my process would be to slowly replace all of the words, take the words and put them in the notes section, and then replace them with images"
Susan	We went to that active learning model in just about all of our curriculum. …. Active learning is really that opportunity to try to engage people on all levels, to try to hit all the modalities of learning, that people can see what you're talking about; they can hear what you're talking about; they can try out what you're talking about.

create a particular result in their project, and she encourages them to do so but does not feel it incumbent upon herself to learn the new technology and teach it to them. She takes the viewpoint that she understands the creative process and the final goal of the class and that she can learn alongside the student if necessary or expect the student to learn independently. She is confident that her expertise lies in understanding and inspiring high quality work rather than in technical manipulation. As a result she has somewhat divorced herself from the constant updating of technology in her classes. She comments that this may not be appropriate in a different type of class, such as programming instruction.

Thus, professors recognize technical change as a reason to modify their courses, and indeed cite this as the primary reason for changing, but seem to gravitate to a preferred teaching style in the process of changing courses. Other research shows that professors will generally teach either according to their own learning style or according to the learning style they were taught in (Felder, 1988; Harb, Terry, Hurt, & Williamson, 1995; Montgomery & Groat, 1998). In this study some of the professors have been influenced by a departmental decision to emphasize a style they call “active learning”. Whatever style they choose, it appears they adapt their classes to that style and then keep it relatively static, while changing other aspects of the course, particularly domain-specific technical content. As will be discussed later, although the style is stable the course itself gets substantially re-designed when content is changed. Part of the problem is that each course is designed as a holistic single piece by a single designer, the professor.

Solo Design vs. Teamwork Design

Another factor of interest in this study was to what extent professors made use of external resources and teamwork to support their design efforts. The short answer is, very little. Most of the professors considered themselves competent in the technologies they were adopting and did not feel the need of support from educational experts or from material creation experts. There were very few exceptions to this in the seven interviews and twenty-plus course change events. Both institutions used for this study boast significant faculty support organizations or learning centers. These learning centers are staffed with professionals trained in instructional design and in creation of teaching materials (graphics, videos, assessment modules and so on). They are freely available to faculty to help with course development. All the participants were aware of these departments but none made any significant use of them.

Some forms of collaboration or teamwork were sometimes used. In the case where a group of faculty was updating several courses from one version of a software package (ActionScript) to a newer version, a few professors collaborated by meeting together to discuss a new technology that they wanted to adopt. They used the collaboration as an opportunity to learn the new technology together but despite the collaboration on technical matters, each professor designed his or her own course independently. Sometimes they look at outside materials but still work alone. Tom stated, "nobody's really helped me design them, but I get input on topics and things from magazines, other professionals that I know that are in the field, or other teachers." In some cases a few of the faculty made use of students to help prepare materials but these students functioned in the role of apprentices, implementing tasks clearly defined by the professor, rather than as collaborators. Adam commented, "I had written the labs up, and I had done most of them. But when I got a TA, I asked the TA to finish doing the labs, and to fix problems that they inevitably would have." Tom also used his TA to give feedback on existing and new course designs and to test the labs. On the whole professors designed course changes alone. As Lisa commented in this context, "It's just me." There was only one exception to this, where David allowed an exceptionally talented undergraduate student to design a series of labs. Even in that case the professor restructured the labs later to fit his personal teaching style and instructional design expectations.

Technology-dependent Content Mixed with Teaching Concepts and Strategy

Several of the professors interviewed try to use the latest ephemeral technology to teach enduring principles. The technical content can affect the teaching approach, and changes in the technical content can impact the design of the course. Susan describes what happened when a new version of software (v. 10) replaced the previous version (v 9.9).

> For example, I used to love in Oracle, in Nine Nine [Ed: version 9.9] you were able to pull out the query tree as a graphic so you could actually see the tree in one of them. It worked for me of course. The most fun thing I liked about that course is the fact that relational algebra is actually there in the database. It isn't just something we made you learn; "Omigosh! there it is." So we take the SQL code and convert it to a relational algebra statement, then we convert it to a tree, and then you could dump the tree right out of Oracle for your SQL – there it is! And then they had all the little notations on the performance on the tree and you could look at that. And then in Ten [Ed: version 10] they dropped that, and it was like so, Grrr! And they went to a text, and then you had to learn to read the text and the indentation and which ones were not indented, and to see what's happening. So now what we have to do is pull out the plan as text and write the tree from the plan, so, Um! you know...

Her expressed frustration ("Grrr!") is because the change in the software from version 9.9 to version 10 means that the aspects she was relying on to convey subtle concepts of relational algebra using a built-in graphic tree structure had been eliminated in the latest software update.

The former version of the database technology (Oracle version 9.9) included a graphic representation of the database structure, which the instructor used as a very powerful graphic illustration of the concept. Students could design their projects and then see the graphical representation of the software structure immediately and directly. The newer version of the software package (version 10) omitted this feature. The instructor was required to re-design her teaching strategy. Students were then required to extract a textual description of the structure and interpret it according to a set of rules in order to generate the same graphical form and thus understand the same concept. Thus the change from one version of the technology to a slightly newer version of the same technology impacted the teaching strategy and led to unexpected and undesirable course redesign. Susan now not only has to create that learning experience herself but the inherent relationship between the technical example and the underlying concept has been broken; she has to artificially try to establish that linkage in the students' minds.

Geoffrey also talked about this phenomenon. He describes looking for "sweetheart applications." The example cited below refers to the design of computer games and how characters in the game world interact with and encapsulate ideas of the software structure (methods and inheritance).

> We say, what is it that we're gonna teach; what is it that they're gonna make, that they're gonna feel good about? And we spend a lot of time trying to find these little sweetheart applications where you can find some application you can build, that is almost like a description of the concept. … A concrete way of expressing an abstraction that is easy for people to get their head around, and then eventually we can move to a more abstract notion of that. But we find that it helps those students a lot to have this very visual, very concrete way of looking at those concepts, and then we say, okay, all of these characters

> need to do those things, but then some special characters are going to have other methods. Oh well, we could inherit those and then we can extend this class to implement these other functions, and that all seems very real to them, very much a part of the kinds of things that they're passionate about. And so it helps for us to embed those concepts, so I say you spend a lot of time in the course design trying to find those nice little apps that have a very nice fit with the concepts that we feel they need to have as outcomes.

In the above example Geoffrey is referring to a technology (ActionScript) that has undergone major changes twice in the last five years. He describes his “sweetheart applications,” which are implementations of the specific technology that encapsulate the concepts that are being taught, in this case object-oriented programming. Such applications contain not only authentic examples of use of the technology but also teach the underlying concepts that enable the students to construct a meaningful understanding of the fundamental principles that undergird the technology. In the above example, since ActionScript has changed twice in the last five years and will probably continue to change at a similar rate, the search for examples in the technology to teach underlying principles must be re-done repeatedly as the technology continues to evolve. In the same vein Jack refers to “fundamental examples” These are technology dependent examples that reveal an underlying educational concept.

Another example occurred when David discussed introducing the new AJAX computer programming methods and language into his course. He referred to the problems the students have in understanding abstract mathematical concepts of computing such as “state machines”, “sequential fetches and steps” and “asynchronous calls”. In particular, asynchronous calls are built into the AJAX language; thus, students programming in that environment will more easily learn that concept as they see it applied in exercises using that language. David describes his experience in integrating underlying principles with current technology.

> I'm really concerned that they get underlying principles of...informing their brain around the way computers work, as far as how to administer them, how to program them, how the state machine model pervades computing...and some people really struggle with that concept of sequential fetches and steps, but those people, if you don't get the basis of how computers do these types of sequences, they'll never be good IT people. So, for me success is anything that builds towards them really getting the underlying "how things work" in computers, and being able to apply that. That's why I said, "Hey, it was kinda exciting to me when I discovered AJAX was based on asynchronous calls." That's an underlying concept that they weren't getting anywhere else. And AJAX probably 10 years down the line will be dead and gone, most likely. But will there be some other technology that has a similar foundation? [It is] likely. And are they going to understand, "Hey, I can have these different servers over here working independently, and they're going to communicate with each other through asynchronous calls! And this one is waiting for a response from this one." This is a good concept for them to know and understand.

Once again David is excited to use a fundamental concept (asynchronous calls) that is embedded in an ephemeral technology (AJAX), and like the other professors, he recognizes that the technology is ephemeral (10 years … dead and gone) and acknowledges that he will have to find and develop some other way to teach that concept when the new technology arrives.

Thus we find that professors recognize the ephemerality of the technology and still tie teaching of fundamental concepts into ephemeral technology. They see learning benefits for the students in linking concepts to authentic learning examples. The inevitable consequence of designing their teaching this way is that they will have to re-design their teaching when the technology inevitably changes.

Apparent Conflict Between Teaching Principles vs. Teaching Technology

Related to the idea of technology-linked instruction is the desire that professors have for their students to understand both the current technology of the discipline and the underlying principles and concepts. All the participants displayed a commitment towards two somewhat contradictory driving forces. On the one hand, they aim to teach fundamental principles or concepts and use the technology to reinforce the teaching of those principles or concepts. They do not teach technology for the sake of technology. Technology is a vehicle to enable the teaching of principles. On the other hand, teaching using the current technology is important. The students expect it and it is necessary in order for them to become competent professionals in the IT discipline. Therefore instructors strive to teach with the current technology. This dichotomy can be seen in this pair of quotations from Susan. On the one hand she says, "I think we are continually running ahead of the snowball. When I teach JAVA, … it seems like every year there's a new version of JAVA that comes out, okay there's new capabilities in that version, so you have to look at it." Later in the interview she states the following:

> We've always felt that IT is an academic discipline. If you're just teaching, "Here's FrontPage, and let's do our web pages with FrontPage," or Dreamweaver or whatever, what you're really doing is you're like an Element K, you're just teaching a software package. We've always tried to stay on the side of, these are the basic IT concepts, principles; this is what we can do. These other things are just our tools.

In the same vein Geoffrey emphasizes the need for using experiential learning to teach core principles when he says technology is, "a concrete way of expressing an abstraction that is easy for people to get their head around, and then eventually we can move to a more abstract notion of that." But in the interview he also states, "With our core languages and so forth, we try and keep very much up to date." Jack recognizes and summarizes this perceived dichotomy when he says, "You can separate outcomes and the technology that gets you to those outcomes, I do believe in that, but you can't ignore what the technology is doing to your curriculum either. You can't completely separate them"

Once again this dichotomy involves participants in frequent re-design. Since they are committed to teaching both fundamental principles and current technology, and in addition they wish to tie the principles and the technology together, they inevitably end up re-designing their courses frequently, as the technology changes.

Management of Change

An issue of significant interest is that, given the environment in which the professors work, where they are frequently changing their courses and where they are not rewarded for it within the institutional system, how do they manage the process of change? This was directly addressed in the interviews and a number of different responses emerged. These responses are shown by the quotations in Table 5.

Here what I term the "heroic conquest" and "formal garden" models of updating becoming apparent. Geoffrey, Jack, David , Tom and Susan are all investing substantial amounts of time to modify courses and also modify course designs (Heroic Conquest). Adam and Lisa are constantly and steadily feeding material into courses that they hold relatively stable (Formal garden). In all cases time investment is significant. These models are discussed in more detail later.

Review of a Single Large Course Change

Although a number of changes in courses were documented it is revealing to discuss a

Table: 5

Different approaches to managing the time and resources required for constantly updating the curriculum.

Alias	Relevant quote(s)
Geoffrey	I find myself trying to see if I can double dip, in other words, can I work on a course of that, I can answer a question about which I might write a paper, and can I design the course so I'll have the data I need to write that paper”
David	[It’s] a labor of love for me, I put in an inordinate amount of work on it, constantly. And partially because it's a very core part of IT, I feel a compulsion to make this an outstanding course.
Jack	What I've tried to do is always change one dimension of the course each iteration. …. Investment in time—substantial. It's what I love doing, and I've been able to, from a faculty standpoint. …. A lot of it during the summer. During the semester, … it's just too difficult to do a lot of it. …. How much effort? Substantial. I would say, if there's a major shift in language, I mean, when we went from 2.0 to 3.0, I was probably putting in about 10 hours a week [Over several weeks]
Tom	Well, It seems that about every course that I've ever taught, I've had to come in and change it, it seems that that's what my role has been. [Tom estimates he spends about 75% of his time on teaching and updating courses. The remaining 25% he spends on research and other responsibilities.]
Adam (1)	I update my lecture notes every time I read an article that is relevant. I usually read during lunch and if I see something that is relevant I'll immediately go right into my lecture notes for next year and just copy this year's lecture notes into next year, change the year, and change them right there. Or if I’m giving a lecture and some student will point something out, I’ll make the note in the margins of my lecture notes and then each year before I teach the class I’ll go back through them and take those notes and incorporate them into the lecture notes. Also if see any articles that I think need to be considered for use, I’ll simply make scans of them, have the secretaries make scans of the articles and I print them out, and then I have both the printed copy and the scanned copy - I have the material to use the next time I teach the class. I just file them in the folder for that class, as well as file them right next to the lecture notebook for that class. So I'm constantly taking that material – of course the next time I teach, I incorporate all that material into there.
Adam (2)	Q?: “About how much time, effort, money, etc. did it take you to create that ECC lab? What was the total investment? Particularly time.” A: “probably six hours.”
Lisa	Q?: Does your class structure stay the same, and you just introduce new material as you find it? L: Yes. And they [the students] send it to me. Like, “Oh, you'll love this!” What they'll do is send me a link and I can just go look at it, and then I put it in the right place. And that's a good way to go.
Susan	There's always a new version of Oracle. I try to stay at least one version back, and not rush towards the new version, but there's a change every single year. I go through every single exercise every single year and I have to run it against whatever my current version of Oracle is, and look to see the changes and what's not visible anymore.

single large change. The change involved updating a single technology, the ActionScript scripting language for programming Adobe Flash animations. This change impacted several courses in the department. As described by the instructor, programming in Adobe Flash has evolved from a simple way to script primitive animation in a web browser, to a fully developed, objected-oriented programming language—ActionScript 3.0. Some of the faculty in the department decided to collaborate in updating their courses to use this new version of the language. Several courses were affected by this change but the effects on a single course that was changed are instructive. The language was updated twice, from version 1.0 to 2.0 and then subsequently from 2.0 to 3.0. In each case a small group of faculty met together regularly in the summer to discuss the new technology and to develop course materials to support it. This involved many hours and experimental development of new labs and programming assignments. In one case a couple of weeks were required to develop a single new lab experience for the students. The faculty were learning the new technology themselves as well as developing new teaching materials using it.

Several benefits were described by two of the professors who participated in this change. Firstly, the students were now learning to work with a fully object-oriented language. This is important because object-oriented languages are the standard in their professional domain and are used elsewhere in the program. In fact the department is now using ActionScript as the introductory programming language for some of the students. Secondly, it is important to the designers that they are using current industry standards. They indicated that they keep in contact with the suppliers of the software and have participated in pre-release testing of systems. In other words, they choose to be on the leading edge of development in this particular field. Because of their close relationship with Adobe, the company that develops and owns ActionScript, updating the language for their classes is an important goal for them. These benefits all relate to the benefit of students using current technologies relevant to their future professional practice. In addition, the adoption of this language as an introductory programming language did lead to some smoothing of the later learning experience for students.

The change was considered very important and consumed many hours of faculty and student time. Since they have done it twice in about five years and have done similar major changes in the past, it can reasonably be surmised that they will go through similar major changes again in the future as the technology changes again.

An interesting factor in this particular change process is that the institution did not actively support the effort invested in the change. The faculty were able to use their free time in the summer to make the change but received no extra funding for the effort, although funding was applied for. Related to this is the expressed opinion of the professors that these efforts are not highly recognized by the institution as valuable. The institution regards these changes as necessary components of the professor's job, but has clearly indicated that they expect the professors to pursue funded research and publication to enjoy future rewards in the form of promotion and salary increases. When one senior professor interviewed during this study indicated to a senior administration official that curriculum updating efforts require considerable time, the response was that faculty were expected to normally work a 70 hour week or more.

This periodic large-effort change in curriculum I have characterized as the "heroic conquest" model. Professors expend great efforts over a relatively short period of time and completely revise the curriculum each time they do it – perhaps during the summer every couple of years. The disadvantages of the heroic conquest model are obvious. In contrast to the heroic conquest model a few professors use what I refer to as a “formal garden” model. They redesign the course to their preferred teaching style and thereafter change domain-specific technical content fairly continuously in small increments, they consciously attempt to minimize changes to the structure of their instructional design. Using the formal garden approach requires considerable discipline by the professor. Although several participants mentioned this approach only one (Adam) seemed to be doing this with any consistency. Adam has one or more pages for every class hour that he teaches and notations on the page indicate specific technology demonstrations or discussions. As he introduces new technologies into his teaching the pages are carefully updated. Other professors (Tom, Lisa) paid lip service to the idea but it was apparent that they were expending considerable effort updating their curricula periodically.

Generally, it requires very large and usually unrewarded expenditures of time and resources to update the curriculum. The advantage is that course content and structure can be rebuilt for new technologies or new teaching methods as necessary. The formal garden model avoids the huge intermittent effort of rebuilding courses but lacks flexibility and adaptability.

Other results emerged from this change that will be discussed in terms of the architectural analysis that follows.

Architectural Analysis

In addition to the recurring themes, the language that the participants used revealed architectural elements of the instructional design. Certain domains or layers of design could be identified in the transcribed interviews. Various design aspects were identified within a set of domains that represent an underlying architectural basis for understanding the process of instructional change. Schön in his analysis of the design process of architects designing a new school building identified a set of “Normative/Descriptive Design Domains,” which were derived from the conversations he observed (1987, p. 59). In an analogous manner I found that professors used words or phrases to indicate that they were focused on a particular aspect or domain of the design process. For example technology product names or descriptions ("Oracle", "ActionScript". "LCD monitors") related to new and changing technology. When professors used these words they were describing instructional design decisions about how to include new technology. Similarly time based words or phrases ("each summer" "start of each semester" "one at a time") reflected design decisions about the pace of change of courses. By collecting similar types of language and seeking common factors in the language a number of design domains were identified. Table 6 shows the design domains that were found.

These design domains not only indicate the multi-faceted nature of the design architecture, they also show other characteristics. Examining each of the domains it appears that each domain represents a certain type of design decision. Parnas, in his discussion of software modularity, indicated that a major purpose of modular design is to enclose design decisions in modules (1972). The different types of decision for each domain were identified and shown in Table 7.

The domains can be classified in more detail. Not only can design decisions be identified, but also the different domains can be further identified in terms of the type of decision or domain that is represented. The Structure-Behavior-Function model (SBF), discussed earlier, also provides further insights into the domains.

Structures, functions and behaviors have been used for analyzing design by multiple designers and theorists in fields ranging from computing, (Clark, Mistree, Rosen, & Allen, 1997; Gero, 1990; Goel & Chandrasekaran, 1989; Goel, et al., 2009; Kruchten, 2005; Schank & Abelson, 1977) to educational design (Collins & Ferguson, 1993; Hmelo-Silver, Holton, &

Table 6

Domains for instructional design with the initial of the relevant participant in parentheses

Domain	Language type	Example words or phrases
Technical content (change)	Technology names or versions	“LCD and CRT Displays,” (A) “ActionScript 3” (J, G), “Oracle” (S), “[version] nine-nine” (S)
	Topics	“Storage devices”, “Displays” (A), “object-oriented paradigm” (J)
Driving force for change	Change verbs or verb phrases	“It's always a changing target” (A), “Market forces and what’s new” (A), “What's hot. We try to stay with the source, pretty much” (S) “industry standards” “market adoption” (D, L), “what's new” (A), “we switched to studio teaching” (D).
Instructional Philosophy	Theory language Names for instructional approaches	“Constructivist” (T, G), “Studio teaching,” (D, G) “Active learning” (G, J, S), “Coaching” (T)
Strategies	Concept, Problem, Interaction, Exercise, Instructional period, challenge, Unit, Lesson	“Basic IT concepts, principles” (S), “Fundamental [idea]” (A, D, G, J), “Assignment” (A, D, T, G, J, L, S), Lab (A, D, T, G, J, L, S), “Project” (A, T, G, J, L), “what is a better way to present this than I've done in the past?” (A), “this is one we will build together in class” (J)
Student interaction (Message/Control)	Explain, Discuss, negotiate, Email, meeting, advise, consult	“Call and Response” (L) “Lots of emails but a few phone calls. And emails are just conversations between you and the student” (A), “I can send emails to them and I can put them in groups” (S), “Clickers keep students involved” (A), “class is discussion of the things that they've read”
Presentation/ representation	Perceivable elements	Images, videos, PowerPoints, present, teach, “I have an image and I talk about the image” (L)
Change management	Resource	“Collaborate”, “team”, “TA”, “resource center
	Management phrases	“The labs themselves didn’t change” (J)

Domain	Language type	Example words or phrases
Data Management	Filenames and names of data organizational systems	“MyCourses” (S) “Blackboard” (A, D), “I just file them in the folder for that class” (A), “I automatically do that [track change history] with file names” (J)
Environmental factors	Organizational and institutional.	“outcomes,” “Program,” “course,” “research, “Lab staff” “diverse group [of faculty]” (G), “[new lab equipment] is a matter of budget always” (L)
	Resources	“funding” (G), “It's just pure volunteer” (J), “we have a chair who's very supportive.” (J) “65 hours per week” (S)
Curriculum structure	Course names or numbers as part of a sequence	“programming sequence”, “prerequisites”, Course numbers (500-level, 350, 628, 4002-330, 4002-434 etc.)
Pace of change	Time-based words	“change one factor at a time” (J), “update at the start of each semester” (several), “do changes in the summer” (J, G)

Table 7.

Decision oriented listing of domains, showing their structural, functional and activity aspects.

Domain	Type	Nature of design decision
Technical Content	S	Professors decide to incorporate new technical content. They do not control the content, only decide to adopt it. This is identified through various specific technical references, such as names or versions of technologies.
Driving force for change	F	Various phrases or words indicate that participants' change decisions were influenced by internal or external forces. The driving force indicates the type of change the professor decides to make.
Instructional Philosophy	F	The professors chose an instructional approach, either theoretically-based or pragmatic.
Strategies [Strategy]	A	Creating instructional strategies, such as class presentations, assignments, discussions, tests etc. Strategies tend to follow the instructional design philosophy.
Student interaction [Message/Control]	A	Student interaction elements of the design are chosen by the professor. Although they may be part of the strategy, they are often more fluid and rapidly changing. Professors may choose different interactions ad-hoc as the learning situation suggests it.
Present/represent [Representation]	A	How the material is presented is a very personal decision. Sometimes it is chosen by the professor in advance, such as prepared demonstrations, PowerPoint slides and graphics carefully collected and included in teaching materials. At other times it may be much more dynamic as a professor spontaneously asks students to participate in a discussion or chooses to present a topic to them or challenges them to find materials to match the subject under discussion.
Change management	F	Change management is a decision relating to how the professors allocate their time to curriculum change, whether they do it as a regulated on-going activity or in bursts of intense effort.
Data Management [Data Management]	F	Data relative to the course must be managed and handled. Most professors see this as an independent function and delegate it to a separate system, such as a learning management system.
Environmental factors	S	The environment that the professors design and teach within. The professors have little control over institutional policies, guidelines and expectations but they choose how their designs will interact with the environment.
Curriculum structure	S	Professors decide to how to include new content within an existing curricular structure in most cases. Changing the curriculum, in the sense of changing sequences of courses is frequently decided at the institutional level. From the point of view of the course designer the curriculum, with its pre-requisites represents a structural element to be designed against.
Pace of change	S	The pace of change is not under control of the professor but they choose how to adapt to it. The decision here is how they choose to synchronize their course changes with the ongoing technological changes and which aspects of their course they allow to be affected by any current change.

Kolodner, 2000; Hmelo-Silver & Pfeffer, 2004). The design domains can be classified in terms analogous to the SBF approach. This not only provides further insight into the design architecture but also reveals relationships between the different aspects of the design. Since the SBF approach has been explored and modified by multiple researchers there are multiple meanings of the terms ‘structure’, ‘behavior’ and ‘function’ as well as similarly named and closely related concepts (such as “form and function”) for describing aspects of design. I have chosen to use the terms Structure (S), Function (F) and Activity (A) and define them as follows.

Designers apply functional intentions (F) to abstract structures of instruction architecture (S) by executing a variety of instructional design activities (A). These structures, functions and activities are different abstract domains or layers of the design process; each domain captures some aspect of design decisions. The usage of these terms will be illustrated by an example.

Technical content changes rapidly and constantly but the professor has no control of that change process and cannot make design decisions affecting that process. Therefore the design decisions relating to technical change relate to how the design will be adapted relative to the fixed reality of change happening. This is therefore a structural element. By the same token the professor has a certain philosophical or intellectual preference for a particular teaching approach, such as “constructivism” or “coaching.” The professor thus indicates a preferred functionality in course design, but this does not define a process of instructional design. This, therefore, is described as a functional domain. To design and create the course the professor designs student learning interactions and selects representations of the material. These are described as activity domains. Another example of a structural domain is the Environment domain, which includes institutional guidelines, course outcomes and similar factors decided at departmental or higher levels within the organization. Once again professors cannot (easily) decide to change these items when designing courses so the design decisions relate to how to adapt the course to these factors. These structural, functional and activity classifications for all of the domains are also included in Table 7.

Finally, in classifying domains, the design layers architectural approach discussed by Gibbons and others also has a close match to some of these design layers. Design layer concepts such as data management, strategy and others emerged as design domains. The well-developed definitions of these design layers, such as the Strategy and Representation layers, provide additional understanding of the implications of these design domains.

Table 6 shows the domains and the type of decision represented in each design domain. It also shows the nature of each domain, whether it is a structural, functional or activity element and therefore how it will relate to other domains. Several of the domains correspond reasonably with known design layers, and in those cases the relevant design layer is shown in square brackets under the name.

The sub-classifications of the design domains, which indicate aspects of the way the domain interacts with other domains, explain much of the process of design that professors follow in these rapidly evolving environments. Professors are constrained to constantly make changes to their courses due to changing technology. This change is a constant factor in information technology. They decide to incorporate the new technologies as they see them becoming essential to the course. What they intend to achieve functionally in terms of the course changes is driven by factors such as “what’s new,” “what’s hot,” or “market adoption.” The types of changes are also driven by other functional intentions, such as by their underlying instructional philosophy, either theoretical or pragmatic. To implement these functional elements they choose various design activities, for example teaching strategies such as assignments, class discussions or specific student interactions, and they choose different representations such as specific images, demonstrations or examples. Professors also follow functional intentions in choosing data management functionalities and change management functionalities. These functions and activities take place against a structural environment of institutional policy, curriculum structure and structural background of pace of change.

There are some special cases among these domain definitions. For example “curriculum”, meaning the high level sequencing of multiple courses to achieve departmental goals, which has been represented here as a structural element, is a structure against which faculty functionally design, choosing to implement curriculum aspects (such as prerequisites) in their design. However, the curriculum is usually defined and designed by the faculty, just as the course content is. The significant difference between curriculum design and course content change is that the curriculum is designed as a separate external process on a much slower time scale. Curriculum change is usually decided by committee and implemented over a long period as approvals wend their way through college and institutional committees, and inter-departmental problems are resolved. Thus, from the point of view of a professor modifying courses to incorporate rapid technological evolution, the curriculum functions as a structural element; from a longer term perspective, curriculum change is another design activity with structural, functional and activity domains of its own, but this is not explored in this research study.

The eleven domains presented in this study are not offered as a definitive set of domains for all applications; they are a set of domains that emerged from the language of the interviews. Some of these domains overlap but they are separated by their relative importance to the designers. For example, the Curriculum domain could be considered as part of the Environmental domain in that they are each part of the institutional background against which the instructor is making changes to courses. However, the curriculum issues play a specific and distinguishable role from the point of view of the instructor modifying a course. The curriculum has a close relationship with course changes, directly affecting prerequisite and post-requisite courses and strongly influencing which topics may be added to or omitted from a course. The Environment domain affects specific design activities more remotely. It impacts decisions relating to credit hours, departmental policies and other considerations, so the designer views these two domains as separate issues, as revealed in the distinguishable types of language used to refer to these concepts in the interviews.

It is entirely possible, even expected, that further studies in this area may lead to different domains or domain boundaries than those elucidated by this case study. Further study on a broader scale will probably permit generalization of the domains to a smaller set or a small set of major domains with subdomains within them. This case study did not provide sufficient data to permit such a generalization, nor clear enough evidence to classify domains as proper subsets of larger domains. Nor was it the intent of this study to achieve such generalization. This study attempted to reveal and describe themes and structures of the design process to gain a better understanding of the influences and results within the research questions. Generalization to the whole field of instructional design can follow from further research.

Implications of Structural Findings

The complexity of the architecture revealed by the design domains is a strong indicator that evolving course design can benefit from modular design thinking as discussed by Baldwin and Clark (2004). The fact that work within some of the identified domains indicate a need for specialized knowledge of some topics (such as instructional philosophy and data management – not typically within the expertise of the professor designing the course) also suggest that this type of design is probably in the region where multiple people are required or recommended for effective design (Baldwin & Clark, 2000).

The differences between structures, functions and activities indicate the nature of design decisions that can be taken by the designer. Design decisions relating to structural elements indicate that these are areas where the designer should not try to change the domain but must include the constraints of this domain in the design. Functional design decisions indicate that the designer should clearly establish preferred modes or directions for design. Design activities indicate where course content should be created or selected to implement the functional modes chosen. Activity decisions reflect where course designers can spend time creating specific instructional artifacts. By recognizing different types of decisions designers can, in the spirit of modular design, create conceptually isolated modules with defined interfaces and reduce the need for complete course re-design when future changes are needed.

Examples of modularization or lack of it, in design domains was seen in the interviews. Baldwin and Clark discuss the necessity of keeping domains separated or the benefits are lost. Susan describes the situation where new technical content, in the form of new versions of the Oracle database software is released every year. She describes how she then has to completely review and re-work her learning activities (lab assignment content) to ensure they are still compatible. The interface between the design domains of technical content and teaching strategies is not defined clearly enough to prevent the redesign of the one because of the other. In contrast to this, Adam describes how he continually inserts new technical content into his courses but keeps the course strategy undisturbed, with much less rework time.

The concepts of modularity, domains and layers of design, as discussed previously, shed further light on the problems of constantly evolving curricula. When reviewing the events described in the interviews it is apparent that although the language shows different domains of design, professors rarely attempted to separate these domains to allow for future changes. The approach to design was somewhat holistic. There were two notable exceptions to this generalization. Firstly, in several cases the officially listed course outcomes were deliberately not changed while the technology of the course was changed. This was partially an attempt to minimize changes within the course and partially an acknowledgement that changing outcomes, which interact with the department's ABET accreditation, generates ripple effect changes at several levels outside the scope of the course. Secondly, the courses were frequently changed to match the professor's preferred instructional philosophy (and strategy) and then (nominally) the strategy was held constant for future changes in technical content. However, these attempts to constrain the disruptive effects of technical change, while well-intentioned, did not achieve their desired effect, as will be shown.

Firstly, the attempt to keep the course outcomes stable shows an inherent awareness of the problem of multiple domains as aspects of design, but in some cases the problem persists and is merely hidden. This was highlighted by the following example from Geoffrey's discussion of updating a programming class. He updated a two-course programming sequence as the programming languages used evolved, most recently moving to an outcomes-oriented language (ActionScript version 2 and then version 3). In his own words he describes attempting to keep the outcomes unchanged. In response to a suggestion that the course was completely redesigned, he says,

> Well, yes and no, because if I was going to tell you this was Programming 1, every time out I'm going to say, We're going to learn about a statement; we're going to learn about a variable; we're going to learn about a function; we're going to learn about a parameter;

> we're going to learn about repetition; we're going to learn about conditional statements. All those things are gonna still be there."

Statements, variables, functions and so forth are indeed programming concepts that will be found in all modern programming languages, including the older procedural languages, from which they are moving away, and are still found in the newer object-oriented languages, which they are adopting. However, programming the object-oriented languages is a quite different experience for the learner, as several researchers have found (White & Sivitanides, 2009; Wiedenbecka, Ramalingama, Sarasammaa, & Corritore, 1999). The effect of these differences is shown when Geoffrey describes his "worst-case scenario."

> Our worst case scenario is the student who comes in, takes Programming One, gets a C, doesn't have a great time, shies away from Programming Two, finds they're a senior and goes back to take it, and not only have they forgotten everything they learned, but the language has taken two quantum jumps since the One – it's very difficult. Our worksheet … says, "you've got the [prerequisite]; you're entitled to take this course," and we know very well that they're not prepared for it. It's a real problem.

So here Geoffrey specifically indicates that the course outcomes did not change, but those course outcomes were defined in terms of low-level technical language definitions, whereas the course changed substantially from one approach to programming style (procedural) to a different style (object-oriented). This considers only the changes in the domain-specific content, not any other changes that may have occurred, but this change alone is sufficient to create significant problems for the learners. The listed course outcomes apparently do not take cognizance of the significant differences between the two programming styles.

Jack was also involved in this change of the programming language relative to other courses that also used ActionScript. He describes the same issue relative to designing computer games involving animated characters, their weapons and fighting capabilities as follows:

> What changed was the technical requirement. Before, you would talk about creating the dynamic functions in one file for doing the different tasks in the game … but now it changed to an object-oriented design, so now we talk about things like manager classes, collections of bad guys, collection of good guys, bullet manager classes for shooting games, how different it is in managing graphical assets when you have an object that represents that, do we create and destroy the object? So the labs themselves didn't change in terms of you're still building a video game, you're still building an interactive greeting card, you're still building an audio application. The underlying technology for getting there changes. The course outcomes never changed. That was the interesting part about it, was the techniques for getting there changed.

Although Jack states that the pre-defined course outcomes never changed in his course ("you're still building a video game") in fact the course changed substantially. The course changed from one type of programming experience to a quite different type of experience moving to an object-oriented language and quite different program design structures, as he describes, but the course outcomes were defined in terms of simple programming constructs and so *apparently* did not change. In this case there is a significant disparity between the nominal (listed) outcomes and the effective, as taught, outcomes of the course that is not being acknowledged, which leads to Geoffrey's "worst-case scenario."

In the second circumstance, professors change the course strategies to suit their own instructional philosophy preferences. Several other layers of the course design were changed at the same time and in such a way that the changed layers were entangled. For example when Lisa changed her courses to her “Call and Response model” (question, challenge and discussion, with exercises – a variation on a studio teaching model), it meant that not only did the strategy layer change but the control and message layers changed too, i.e. the strategy of presenting material to the students was predicated around a set of ideas that were to challenge student thinking. The control layer mechanisms whereby students would react to these ideas were closely tied to the strategy, and the message layer mechanisms for conducting learning conversations with the students were embedded in the strategy. Furthermore, the representation of the material presented to the students (representation layer) was specifically adapted to the strategy layer. If the same instructor teaches the same course with changes only within the specific content this will work for a while, but when a different instructor teaches the course or when changes are made that affect any of the layers described, the associated layers will also be disturbed. Large sections of the course will have to be re-designed again. Lisa acknowledged this problem by indicating that when different colleagues teach alternate sections of this course they share materials and ideas but each of them has to customize the course design to suit their own approach. She describes it as follows:

> There are like nine sections of that course. And there've been times when there have been four or five of us teaching the course in a particular quarter. And what will happen is we trade materials. So Jane will “Janeify” my thing, I will “Lisafy” my things, things will be “Charliefied”. And they end up being shared. And we've all been around enough that we're comfortable taking our own particular spins. One professor is very structured and has their class exercises—do A, B, C, D in this order. Mine is, “I want you to create this, how you get there I'm not sure, I'll show you how if you need it.” [from participant “Lisa” Edited from the transcript to anonymize names and for clarity.]

Using the analogy of Brand's building layers presented earlier, this is analogous to saying that when we install dishwashers in homes each homeowner has to rebuild the dishwasher to work with different power voltages, different plumbing fittings and different space constraints. While this leads to very individualistic designs it is very inefficient and leads to haphazard changes to the system as well as poor transferability of course designs.

This set of design domains also provides additional understanding for the "worst case" example cited earlier. The designers chose to create new learning activities (Student interactions and representations) against the technical content structure (new version of ActionScript software). Unfortunately they only paid lip service to the Environment design domain in that the outcomes were assumed to be static. They did not evaluate the degree of change in course strategy; nor the problem of activities not being coordinated with the outcomes. Secondly, they also did not consider the Curriculum structure in their design functions and activities and ended up with a new course that was disassociated from some of its essential domains. This disconnect resulted in a poor learning experience for several students. Revealing the design domains identified both problem causes and potential solutions.

Research Evidence Addressing the Research Questions

Each of the research questions and sub-questions is listed and evidence from the research results is used to address the questions.

Question one is, "What is the nature of the changes that occur in instructional design architecture when courses in higher-education Information Technology evolve?" This question is addressed through several sub-questions. Each of the sub-questions for question one will be addressed below.

What effect do changes have on the instructional design architecture? With the instructional architecture shown in tables 5 and 6, the changes in technology occur in the technical content domain. However, instructors allow those changes to drive changes into other domains, including strategy, student interactions and presentation, that cause the extensive redesign efforts that were reported. Faculty are aware of change, and the language-derived domain analysis shows that they compartmentalize it to some extent, but although they seem to differentiate between strategy and content and aim to address one or the other, they inadvertently impact many others. For example, in the ActionScript 3 updating experience the intent was to change the domain-specific technical content; they deliberately intended to keep the course outcomes unchanged. When they adopted the new technical content they effectively changed parts of the design in several domains (labs, class presentations, assignments). The prevailing teaching philosophy ("Active learning") was still used. However, they changed one of the key things they wanted to keep constant, the effective course outcomes (not the listed course outcomes) – and it negatively impacted the learning experience of at least some of the students, as discussed in the "worst case scenario".

There are both intended and unintended effects. The latter outweigh the former. ABET has a significant influence here. Both the universities used for survey data have their programs accredited by ABET, which requires both defined course outcomes and continuously measured performance assessment and improvement. To work within this environment institutions define course outcomes, which are related to program outcomes. Changing these requires changing the departmental assessment structure—a complex process that is avoided. Therefore, at least to some extent, the course outcomes are defined in conceptual terms, allowing instructors the freedom to modify technical course content (domain specific content) to keep up with changing technology, while satisfying the same pre-defined course outcomes.

How frequently do changes occur? Two different approaches were found to dominate the timing of changes. Changes in technology are continuous; changes in courses are graduated. Minor changes are "slipstreamed" directly into the courses but major changes are regulated by the normal academic time flow, i.e. either by semester or by academic year—with new designs being developed in the summer, as dictated by external events (e.g. updating a major computer software package such as SQL or ActionScript). Generally, it appears faculty allow the normal academic timetable to define the timing of changes. For major changes the summer break plays a role.

What are the types of changes that occur (content, procedures, equipment)? Observed changes occurred extensively in labs. However, it appears that changes in technology also almost inevitably led to changes in class presentation material. Lab equipment changes (frequently software) also happened on a regular basis.

What is the range of extent and granularity of the changes? Changes were found across the entire range of granularity discussed, i.e. there were minor changes, like a new image for a class presentation, and there were new course designs and multi-course designs.

What motivates the changes? As expected, changes were motivated by the perceived need to teach with the latest technology. Professors described their motivations using phrases such as "what's hot" and "market adoption." However, instructors also made changes based on their desire to adopt a particular teaching philosophy (e.g. active learning or studio teaching). As indicated previously change in technology was sufficient motivation by itself for course modification or re-design but changes in teaching style were always accompanied by technology changes in this study. This is possibly because technology changes rapidly enough that a change in technology can be undertaken anytime a course is being altered for other reasons. Finally student feedback and perceived student need also led directly to course changes.

How are the changes managed, i.e. what models or templates (deliberate or unconscious) are used to manage the change process? Two models were noted one of which was dubbed the “heroic conquest” model, where considerable effort and sacrifice of time by the professors leads to a completely re-designed course with new technology, new assignments, new ways of teaching fundamental concepts and new equipment (software). The second model is the “formal garden” model where the structure is defined and laid out and new technical content is dropped into place as seems appropriate. Although several professors talked about this strategy, only one seems to be implementing it (Adam). The others make statements like “What I've tried to do is always change one dimension of the course each iteration. Not to try to throw the kitchen sink at it, but OK, here's one thing we can adjust this time” but in fact make major changes as described by the heroic conquest model.

The second research question relates to how the instructors interact with the process of change. The second research question was, “what actions or processes do instructors implement when changing courses?” The five sub-questions of research question two will be addressed below.

What actions do instructors take in order to keep to keep their workload manageable? A number of different strategies were described here, ranging from “double dipping” – trying to use the results of instructional re-designs as part of their scholarly research, to using not the latest but the next-to-latest technology (“I try to stay at least one version back.”). Lisa talks about deliberately not pursuing the latest technology but staying committed to excellent design in her discipline and letting others (students, lab managers) explore the technology first. Another strategy, if it can be called that, is just to work longer hours.

To what extent is curriculum change a solo effort and to what extent is it shared or delegated? With very few and limited exceptions curriculum change is a solo effort by the instructor. Supporting resources are available in the form of instructional design departments but they are not used. Collaboration is used to understand the new technology but sharing of materials is severely hampered by the widely divergent and largely incompatible strategies and teaching philosophies that instructors choose to use.

What do instructors do to ensure the students' educational experience isn't harmed by updating the curriculum? The apparent quality control mechanism is the good intentions of the professors. Instructors express a desire to improve the educational experience of students. The attitude is more one of benefiting the students, rather than of holding the curriculum harmless. There is an assumption that changing to the latest technology is the best way to benefit the students. There is little evidence that the instructors, when they design their new courses or modules, explicitly consider or evaluate the impact of the changes on the quality of the educational experience of the students. The exception is the example of holding the outcomes constant for a course; however, this was motivated as much by the bureaucratic difficulties of changing them, combined with the naïve belief that they were based on foundational concepts that accurately described the course, as it was on any expressed desire to protect the students' educational experience.

What principles do course designers/instructors use in deciding what to change and what to keep? With reference to the architecture model, instructors do not change items in the Environment and Curriculum domains, nor do they (often) change their teaching philosophy. They do change the technical content, and with the technical content they change their strategies, student interactions and representations. Their data management domain seems to be reasonably static. This is partially because some instructors choose to use the institutionally provided data management systems (e.g. Blackboard LMS), but those instructors that have their own systems, such as naming conventions for their files on their computer, keep to those systems too.

What choices (trade-offs) do instructors make when modifying the curriculum? In several instances professors tried to change the technical domain-specific content without changing the course design. This was not always successful; in fact in most instances studied it was not successful. When technical content changes the professors in this study discarded learning modules that were tied to outdated technical content and created new (different) modules that matched the new technical content. In doing so they discarded prepared examples and learning experiences tied to the technical content, such as Susan's concern for losing the cognitive learning benefits of a graphical display of database structure in upgrading from one version of the software to the next. Thus the professors sacrificed prepared learning material in order to introduce newer technologies and either had to re-create the learning experiences or forgo them.

The constant focus of introducing new material into the curriculum comes at a cost of carefully designed learning experiences and also at a risk of impacting their promotion and tenure within the institution. They had to sacrifice personal time to keep up with the constant changes required while still trying to satisfy the other expectations of their institutions. As a secondary effect, although they strive to meet their students' needs, by focusing on technical content they may pay less attention to student learning. This was evidenced not only by the motivations for change described by the respondents (primarily latest technology) but also by the study done by Lister and Box (2008), which indicated the emphasis in published computer educational research on technical change rather than learning development.

Summary of responses to research questions. All of the research questions were addressed in the study. Having thus considered the issues and analyzed the data certain conclusions follow. These conclusions and the suggested actions that follow from them are presented and discussed in the following chapter.

Chapter 5: Conclusions and Future Research

This case study revealed many characteristics of how the selected Information Technology instructors continually re-design their courses and the design architectural implications of that evolutionary process. Some expectations were confirmed and some new findings emerged. A model based on domains of design was found by studying keywords and themes in the data. This model describes the structure or architecture of the design process. The benefits of this model to developing solutions to the problems of rapidly changing IT curricula are discussed.

Recurring Themes in the Data

A qualitative inquiry approach was used to evaluate the interviews with the participants. From this analysis themes emerged that were common among most, if not all, the participants. These recurring themes established a foundation for understanding the phenomena of frequent course evolution in Information Technology.

Firstly, the domain-specific technology is indeed changing continuously and substantively and from the interviews it is clear that this does lead directly to continuous and substantive changes in IT courses. The professors identified the change in technology as a primary motivator in changing their courses frequently. Their personal instructional philosophy was another important factor motivating course changes. In addition the analysis revealed that course change design is largely a solo activity. The professors may consult with colleagues about the technology and sometimes use teaching assistants to test some assignments, but they design the courses alone and, with very few exceptions, did not collaborate in instructional design or use the available on-campus instructional design resources.

As for how faculty manage the change process, a spectrum of different methods was found, ranging from intensive multi-week efforts to completely re-design whole courses, through regularly reviewing and changing sub-sections of courses (particularly labs), down to incrementally and steadily editing course materials. Two different models of change-management are offered to characterize the extremes of this range. The first is named the "heroic conquest" model. This model is characterized by periodic bursts of intense effort to adopt a new technology or instructional philosophy and includes major course re-design. In this model instructors learn the new technology or instructional approach, sometimes alone and sometimes with colleagues. They then discard most, if not all, of the old course material and design new materials, complete with designed student interactions, laboratory assignments, strategies and presentation materials. The alternative model is named the "formal garden" model which lays out the course design neatly and then keeps it rigidly pruned and aligned, with new technical content permitted only under carefully controlled circumstances. New technical content replaces old, but is inserted at the same point in the class materials and old labs evolve carefully, step by step into newer ones. This model requires a small but steady effort rather than large bursts of energy, but at the price of possibly stifling innovation in instructional design within the rigid barriers of the formal garden structure.

When the professors change the courses they do not explicitly address the issue of ensuring that the changes do not harm the educational experience of the students. Course design is oriented towards technical content and thus introducing new content was sufficient justification for change and was also the basis for how changes were to be made. Although they did not directly address the issue of educational impact they seemed sure that introducing new technologies would be beneficial for the students' future professional success.

In summary, the recurring themes showed that professors in the study accept the need to continually incorporate changes into their curriculum. The professors are willing, and in several cases eager, to stay current in their discipline and to keep their courses current, but they find that continually modifying their curricula, while necessary and expected, is nonetheless resource-consuming and unrewarded. They also confirmed that their institutions do not strongly value these course design and re-design efforts in terms of promotion or financial rewards, or even financial support for extra time spent. The re-design approaches succeed in developing new courses but suffer from significant weaknesses.

Proposed Design Architecture Model

This research sought to discover an underlying structural architecture for the instructional design process of this case study. Gibbons, Reigeluth and others have described design layers and design architectures, and determined that separate design languages and terminology are used on each layer (Gibbons & Rogers, 2009; Reigeluth & Carr-Chellman, 2009). An approach similar to Schön's (1987) was used to find domains of design in the data. The design domains were identified by the different terminology that participants use when describing different aspects of the design process. Using this approach a set of design domains was found and described. The ideas presented here synthesize and extend concepts presented by these researchers and also ideas presented by researchers in the field of design, such as Blaaw, Brooks, Parnas and others (Blaauw & Brooks, 1997; Brooks, 2010; David L. Parnas, 1972; D. L. Parnas, 1979).

Parnas and other researchers in the field of design suggest that design is a process of making a series of decisions and that each separate design domain or module encapsulates a design decision. The types of decision that the design domains encapsulate were also found and described. Separating decisions and modules is a key step in being able to develop different parts of the design separately, and thus gain all the benefits of modular design (Baldwin & Clark, 2004).

The domains in this study were defined by identifying design decisions and common elements in the language used by interviewees. This process parallels that of modular design and suggests that the design domains identified could be used to develop an approach to instructional design. Baldwin and Clark (2004) describe this part of the design process in terms of complexity and abstractions.

> A complex system can be managed by dividing it up into smaller pieces and looking at each one separately. When the complexity of one of the elements crosses a certain threshold, that complexity can be isolated by defining a separate abstraction that has a simple interface. The abstraction hides the complexity of the element; the interface indicates how the element interacts with the larger system. (p64)

This suggests that designers can use the results of this study to define useful abstractions (domains) in developing new courses. Combining the ideas of Parnas and those of Baldwin and Clark, each of these abstractions should encapsulate a design decision or a certain level of complexity.

The design domains were analyzed further using concepts of Structures-Behaviors-Functions (SBF), and related ideas (Clark, et al., 1997; Collins & Ferguson, 1993; Goel & Chandrasekaran, 1989; Goel, et al., 2009; Hmelo-Silver, et al., 2000; Hmelo-Silver & Pfeffer, 2004). The form of SBF used in this study described structures, functions and activities. Each of the design domains was classified accordingly.

Having developed a design architecture description from the data, a complex design structure was revealed. The complexity, as described by Baldwin and Clark, (2004) indicates that the design process would be benefited by modular design and collaboration amongst multiple designers. Modular design leads to ease of changing (updating) systems and the opportunity to experiment with parts of the system, with minimal disturbance to other parts (Baldwin & Clark, 2004; Brooks, 2010; Garud & Kumaraswamy, 1995; D. L. Parnas, 1979). This is of obvious benefit to a rapidly-changing curriculum, such as is described in this study.

This research study extends the ideas of design layers and design domains and suggests a structure of abstraction that includes the multi-faceted idea of dividing the design space into separable entities defined not only by decision, domain type and complexity, but also by whether the design issue represents structural, functional or activity issues, using the SBA analysis. Although this study did not create nor recommend a design methodology, the structure implies that design methodologies based on complexity, abstraction and decision encapsulation may be adapted and applied to instructional design as applied to managing constant change in Information Technology course design.

The design domain architecture helped to explain some of the observed effects in the data. Design domains should encapsulate design decisions. When design decisions get spread among multiple domains the design problem becomes more difficult. For example, the professors found difficulties in teaching fundamental concepts using the latest technology. They were trying to satisfy conflicting constraints, and these constraints impacted different domains in the domain architecture model. Firstly, they wanted to teach the latest technology (Technical content domain); secondly, they wanted to teach fundamental concepts, not just the latest technology (Strategy domain); thirdly, they wanted to teach the concepts by embedding them in the latest technology (both strategy and presentation domains); fourthly, they needed to change the technology often (pace of change domain). As a result, changing technology led to designing changes in multiple domains and this translated to extensive alterations to many parts of the courses. These multi-domain design changes led to significant frustration (Susan's comment was "Grrr!") and to considerable extra work in redesigning courses.

Another example of this problem was found in what was described as the "worst-case" example. The professors were changing the course for new technical content (technical domain) and consciously avoiding changing the course outcomes (environment domain). Unfortunately, the design decisions for the course outcomes were poorly defined and were not clearly encapsulated in the environment domain, such that design decisions relating to programming style were handled in the strategy and content domains when they should have been handled in the environment domain. This inappropriate splitting of the environment domain decisions led to the situation where students tried to take later courses and found themselves unprepared, as described earlier.

Sometimes faculty did contain design decisions within a domain, for example data management. Faculty made decisions about managing their data, which may have changed periodically, but did so independently of other design decisions. Where changes were confined to a single domain in this manner there was far less disruption and extra work when courses evolved.

Future Research and Development

This study laid a foundation for future research and development of solutions to the problems associated with continuous and rapidly changing technical content of IT curricula. Several areas of future research are recommended to continue this work. Immediate applications of this work are also proposed.

One proposed future activity in this research area is to share the results with instructors in the field who are actively developing IT courses. Revealing the underlying architecture of the instructional design process and similarities to effective design in other domains of design in this case study provides valuable information for instructional designers. Disseminating this work among instructional designers in the field of computing education design and inviting them to look for similarities in their own applications of design is expected to highlight the problems that changing aspects of multiple layers of design have on the effort and effects of changing IT courses. It is not unreasonable to expect that an awareness of the layered nature of instructional design among IT faculty may lead to more efficient design processes and methodologies and also a useful understanding of the probable effects of design change on the course structure, on teaching strategies and on student learning. It is also expected and hoped that the resulting conversation amongst IT instructional designers will test and modify the design domain model as it is interpreted and applied in other applications beyond this case study.

As a future development in this area, instructional design methodologies need to be sought or developed which will accommodate rapid course change. Such a methodology should accommodate design architecture domains and layers, as an awareness of domains can both clarify the issue of different rates of aging and the principles of modularity.

Furthermore, selecting or developing an appropriate methodology requires an awareness of early constraints in the design process. Research shows that design, including instructional design, is a process of making a series of design decisions. Early design decisions can be made starting from many different points. Designers rationally start with the most important issues (regardless of what process they use to assign importance). Each succeeding decision is then constrained by the earlier decisions. These decisions may be represented as a decision tree (Blaauw & Brooks, 1997; Brooks, 1975). In developing future solutions to the problem of constantly changing IT courses some factors must be prominent early in the design process. From this research, both technical content and instructional philosophy must be allowed to change. Extending this research requires a deeper understanding of design decisions. Design decisions are made while developing courses, while adapting courses to changing needs and even while delivering courses. This is particularly true for IT faculty who act as solo developers in all phases of this process. These design decisions need to evolve into course changes using a coherent process to ensure both effective learning for students and efficient use of faculty time. Parnas was influential in revolutionizing software design through his emphasis on encapsulating design decisions while designing software. In an analogous manner, future research in the field of instructional design can similarly explore the process of decision-making and the types of decisions being made in different design domains to enable similar encapsulation of instructional design decisions with, hopefully, similar benefits for the modular design of computing courses.

As noted earlier, within each of the architectural domains different types of design activity may take place. Researchers have discussed the need for design languages in each domain (Botturi, Derntl, Boot, & Figl, 2006; Gibbons & Rogers, 2009). If development or adoption of design methodologies includes consideration for using different types of skills in different domains then IT instructors are more likely to recognize the benefits of collaboration with instructional design and other specialists in the course design. Future research needs to address the range of skills required for successful design of changing IT courses and relate those skills to the skill-set of typical IT instructors.

One of the research questions asked what measures the professors used to ensure that the students' educational experience was not harmed. The findings were that the instructors make no deliberate efforts to achieve this goal. Future research must include some measures or indicators to ensure that this ensues. IT programs that choose to accredit with ABET are expected to demonstrate a program of continuous improvement (ABET, 2008) so future research into methodologies can relate to mechanisms that schools use in this regard.

In conclusion, this study has found a number of useful models for understanding the process of design in rapidly-changing environments. These models help to explain the problems observed in course changes and point to future solutions to these problems.

References

Criteria for Accrediting Computing Programs: Effective for Evaluations During the 2008-2009 Accreditation Cycle (2007).

Criteria for Accrediting Computing Programs: Effective for Evaluations During the 2009-2010 Accreditation Cycle (2008).

Alexander, C. (1964). *Notes on the synthesis of form*. Cambridge,: Harvard University Press.

Alexander, C. (1979). *The Timeless Way of Building* (Vol. 1): Oxford University Press.

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Construction* (3 ed. Vol. 2). New York: Oxford University Press.

Alford, K. L., Carver, C. A., Ressler, E. K., & Reynolds, C. W. (2004). *A curriculum framework for evolving an information technology program.* Paper presented at the 34th Annual Frontiers in Education: Expanding Educational Opportunities Through Partnerships and Distance Learning, Savannah, GA, USA.

Alkin, M. C. (2004). *Evaluation roots : tracing theorists' views and influences*. Thousand Oaks, Calif.: Sage Publications.

Allert, H., Dhraief, H., & W., N. (2002). *Meta-Level Category Role' in Metadata Standards for Learning: Instructional Roles and Instructional Qualities of Learning Objects.* Paper presented at the COSIGN-2002, University of Augsburg, Germany.

Baldwin, C. Y., & Clark, K. B. (2000). *Design Rules, Vol. 1: The Power of Modularity* (1 ed.): The MIT Press.

Baldwin, C. Y., & Clark, K. B. (2004). Modularity in the Design of Complex Engineering Systems.

Blaauw, G. A., & Brooks, F. P. (1997). *Computer architecture: Concepts and Evolution*: Addison wesley.

Botturi, L., Derntl, M., Boot, E. W., & Figl, K. (2006, 2006). *A Classification Framework for Educational Modeling Languages in Instructional Design.* Paper presented at the Sixth International Conference on Advanced Learning Tehcnologies (ICALT), Los Alamitos, CA.

Brand, S. (1994). *How Buildings Learn: What Happens After They're Built*. London: Viking Penguin.

Brock, D. C., & Moore, G. E. (2006). *Understanding Moore's law : four decades of innovation*. Philadelphia, Pa.: Chemical Heritage Foundation.

Brooks, F. P. (2010). *The Design of Design* (1 ed.): Addison Wesley, Pearson education.

Clark, D. D., Mistree, F., Rosen, D. W., & Allen, J. K. (1997, June 1997). *Function-Behavior-Structure: A Model for Decision-Based Product Realization.* Paper presented at the ASEE Annual Conference & Exposition, Milwaukee.

Collins, A., & Ferguson, W. (1993). Epistemic forms and Epistemic Games: Structures and Strategies to Guide Inquiry. *Educational Psychologist, 28*(1), 25-42.

Computing Curricula 2001: Computer Science. (2001). Retrieved from http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf.

Computing Curricula 2005: The Overview Report. (2005). ACM, AIS, IEEE-CS.

Corbin, J., & Strauss, A. (2007). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (3 ed.): Sage Publications.

Denzin, N. K. e., & Lincoln, Y. S. e. (2000). *Handbook of Qualitative Research* (2 ed.). Thousand Oaks, California: Sage Publications.

Douglas, I. (2001, 2001). *Instructional Design Based on Reusable Learning objects: Applying Lessons of Object-Oriented Software Engineering to Learning Systems Design.* Paper presented at the Frontiers in Education 2001, Reno, NV.

Edelson, D. C. (2002). Design Research: What We Learn When We Engage in Design. *Journal of the Learning Sciences, 11*(1), 105-121.

Ekstrom, J. J., & Lunt , B. M. (2003, 2003). *Education at the seams: preparing students to stitch systems together; curriculum and issues for 4-year IT programs.* Paper presented at the Conference On Information Technology Education (formerly CITC), Proceedings of the 4th conference on Information technology curriculum, Lafayette, Indiana, USA.

Felder, R. M. (1988). Learning and Teaching Styles In Engineering Education (updated 2002). *Engineering Education, 78*(7), 674-681.

Fiscus, J. G., Ajot, J., & Garofolo, J. S. (2007). The Rich Transcription 2007 Meeting Recognition Evaluation.

Fontana, A., & Frey, J. H. (2000). The Interview: From Structured Questions to Negotiated Text. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research* (Second ed., pp. 645-672). Thousand Oaks, London, New Delhi: Sage Publications Inc.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1993). Design Patterns: Abstraction and Reuse of Object-Oriented Design. In O. Nierstrasz (Ed.), *ECOOP' 93 — Object-Oriented Programming* (Vol. 707, pp. 406-431): Springer Berlin / Heidelberg.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software* (1st ed.): Addison-Wesley Professional.

Garud, R., & Kumaraswamy, A. (1995). Technological and organizational designs for realizing economies of substitution. *Strategic Management Journal, 16*(S1), 93-109.

Gero, J. S. (1990). Design prototypes: a knowledge representation schema for design. *AI Mag., 11*(4), 26-36.

Gibbons, A. S. (2000). *What and How do Designers Design? A Theory of Design Structure*. Paper presented at the AECT 2000.

Gibbons, A. S. (2009, May 7–8, 2009). *A Theory-Based Alternative for the Design of Instruction: Functional Design.* Paper presented at the DESRIST '09, Malvern, PA, US.

Gibbons, A. S., & Bunderson, C. V. (2005). Explore, Explain, Design. In K. Kempf-Leonard (Ed.), *Encyclopedia of Social Measurement* (1 ed., Vol. 1, pp. 927-938). New York: Elsevier.

Gibbons, A. S., & Rogers, P. C. (2009). The Architecture of Instructional Theory *Instructional Design Theories and Models: Building a Common Knowledge Base (Vol III)* (Vol. III, pp. 305-326): Routledge.

Glaser, B. G., & Strauss, A. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Hawthorne, NY: Aldine de Gruyter.

Goel, A., & Chandrasekaran, B. (1989). *Functional representation of designs and redesign problem solving*. Paper presented at the Proceedings of the 11th international joint conference on Artificial intelligence - Volume 2.

Goel, A. K., Rugaber, S., & Vattam, S. (2009). Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 23*, 23-35.

Gorka, S., Miller, J. R., & Howe, B. J. (2007). *Developing realistic capstone projects in conjunction with industry*. Paper presented at the Proceedings of the 8th ACM SIGITE conference on Information technology education.

Guba, E. G., & Lincoln, Y. S. (1989). *Fourth Generation Evaluation*. Newbury Park, London, New Delhi: Sage.

Harb, J. N., Terry, R. E., Hurt, P. K., & Williamson, K. J. (1995). Teaching through the cycle: application of learning style theory to engineering *BYU Press*.

Hmelo-Silver, C. E., Holton, D. L., & Kolodner, J. L. (2000). Designing to Learn About Complex Systems. *Journal of the Learning Sciences, 9*(3), 247 - 298.

Hmelo-Silver, C. E., & Pfeffer, M. G. (2004). Comparing expert and novice understanding of a complex system from the perspective of structures, behaviors, and functions *Cognitive Science, 28*(1), 127-138.

Jacobson, M., & Wilensky, U. (2006). Complex Systems in Education: Scientific and Educational Importance and Implications for the Learning Sciences. *Journal of the Learning Sciences, 15*(1), 11-34.

Jamieson, L. H., Edwardson, J. A., & Lohmann, J. R. L. (2009). *Creating a Culture for Scholarly and Systematic Innovation in Engineering Education: Ensuring U.S. engineering has the right people with the right talent for a global society*: American Society of Engineering Educators (ASEE).

Janesick, V. J. (2000). The Choreography of Qualitative Research Design: Minuets, Improvisations and Crystallization. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research* (pp. 379-399). Thousand Oaks, California: Sage Publications.

Joint Committee on Standards for Educational Evaluation., & Sanders, J. R. C. (1994). *The program evaluation standards : how to assess evaluations of educational programs* (2nd ed.). Thousand Oaks [Calif.]: Sage Publications.

Kolb, D. A. (1984). *Experiential learning : experience as the source of learning and development*. Englewood Cliffs, N.J.: Prentice-Hall.

Kruchten, P. (2005). Casting Software Design in the Function-Behavior-Structure Framework. *IEEE SOFTWARE*(March/April 2005), 52-58.

Lamancusa, J. S., Zayas, J. L., Soyster, A. L., Morell, L., & Jorgensen, J. (2008). The Learning Factory: Industry-Partnered Active Learning. *Journal of Engineering Education*(January 2008), 7.

Lidtke, D. K. (1998). *What's new in curriculum design: working with industry.* Paper presented at the 28th Annual Frontiers in Education Conference.

Lincoln, Y. S., & Guba, E. G. (2000). Paradigmatic Controversies, Contradictions and Emerging Confluences. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research* (2 ed., pp. 163-188). Thousand Oaks, London, New Delhi Sage Publications Inc.

Lindsey, L., & Berger, N. (2009). Experiential Approach to Instruction. In C. M. Reigeluth (Ed.), (Vol. III, pp. 117-142): Routledge.

Lister, R., & Box, I. (2008). *A citation analysis of the SIGCSE 2007 proceedings*. Paper presented at the Proceedings of the 39th SIGCSE technical symposium on Computer science education.

Lunt , B. M., Ekstrom, J. J., Gorka, S., Hislop, G., Kamali, R., Lawson, E. A., et al. (2008). *Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology*: ACM, IEEE-CS.

Lunt, B. M., Ekstrom, J. J., Lawson, E. A., Kamali, R., Gorka, S., & Reichgelt, H. (2004, June, 2004). *Defining the IT Curriculum: The Results of the Past 2½ Years* Paper presented at the American Society for Engineering Education Annual Conference & Exposition: Engineering Education Reaches New Heights, Salt Lake City.

Lunt, B. M., Lawson, E. A., Goodman, G., & Helps, C. R. G. (2002, June, 2002). *Designing an IT Curriculum: The Results of the First CITC Conference* Paper presented at the American Society for Engineering Education Annual Conference & Exposition, Montreal.

Machanick, P. (2003). Principles versus artifacts in computer science curriculum design. *Comput. Educ., 41*(2), 191-201.

McGreal, R. (2004). Learning Objects: A Practical Definition. *International Journal of Instructional Technology and Distance Learning, 1*(9), 21-32.

Merrill, M. D. (2000). Knowledge Objects and Mental-Models *The Instructional Use of Learning Objects*.

Molebash, P. E. (2000). Tomorrow May Bring: Trends in Technology and Education.

Mollick, E. (2006). Establishing Moore's Law. *Annals of the History of Computing, IEEE, 28*(3), 62-75.

Montgomery, S. M., & Groat, L. N. (1998). Student Learning Styles and Their Implications for Teaching. *University of Michigan: CRLT Occasional Papers*(10).

Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics, 38*(8).

Pallett , D. S. (2003, Nov 2003). *A look at NIST'S benchmark ASR tests: past, present, and future*. Paper presented at the 2003 IEEE Workshop on Automatic Speech Recognition and Understanding

Parnas, D. L. (1972). On the Criteria To Be Used in Decomposing Systems into Modules. *Communications of the ACM, 15*(12), 1053-1058.

Parnas, D. L. (1979). Designing Software for Ease of Extension and Contraction. *Software Engineering, IEEE Transactions on, SE-5*(2), 128-138.

Patton, M. Q. (2002). *Qualitative research and evaluation methods*: Sage.

Pickard, A., & Dixon, P. (2004). The applicability of constructivist user studies: How can constructivist inquiry inform service providers and systems designers? *Information Research, 9*(3 paper 175).

Proceedings of the 9th ACM SIGITE conference on Information technology education. (2008). Cincinnati, OH, USA.

Reichgelt, H., Lunt , B. M., Ashford, T., Phelps, A., Slazinski, E., & Willis, C. (2004). A Comparison of Baccalaureate Programs in Information Technology with Baccalaureate Programs in Computer Science and Information Systems. *Journal of Information Technology Education (JITE), 3*(2004), 19-34.

Reigeluth, C. M. (1999). 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* (Vol. II, pp. 5-29): Lawrence Erlbaum Associates.

Reigeluth, C. M., & Carr-Chellman, A. A. (Eds.). (2009). *Instructional-design theories and models: Building a Common Knowledge Base (Vol III)* (Vol. II). New York and London: Routledge: Taylor and Francis.

Roberts, L. G. (2000). Beyond Moore's law: Internet growth trends. *Computer, 33*(1), 117-119.

Robson, R. (2002, Oct 2002). Reusable Learning Objects. *e-learning Magazine, Vol. 3* 18 - 19.

Ryan, G. W., & Bernard, R. H. (2000). Data Management and Analysis Methods. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research* (pp. 769-802). Thousand Oaks, California: Sage Publications.

Ryan, G. W., & Bernard, R. H. (2003). Techniques to Identify Themes in Qualitative Data. Retrieved from http://www.analytictech.com/mb870/Readings/ryan-bernard_techniques_to_identify_themes_in.htm

Schaller, R. R. (1997). Moore's Law: Past, Present and Future. *IEEE Spectrum, 34*(6), 52-59.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*: Psychology Press.

Schön, D. A. (1987). *Educating the reflective practitioner: toward a new design for teaching and learning in the professions* (1st ed.). San Francisco: Jossey-Bass.

Shalloway, A., & Trott, J. (2002). *Design patterns explained : a new perspective on object-oriented design*. Boston, Ma.: Addison-Wesley.

Smith, M. K. (1996, 2000). Curriculum theory and practice. *the encyclopaedia of informal education*. Retrieved from http://www.infed.org/biblio/b-curric.htm

Smith, M. K. (2005). Elliot W. Eisner, connoisseurship, criticism and the art of education. *the* *encyclopaedia of informal education* Retrieved 3/4/2009, 2009, from www.infed.org/thinkers/eisner.htm

Sosteric, M., & Hesemeier, S. (2002). When is a Learning Object not an Object: A first step towards a theory of learning objects. *International Review of Research in Open and Distance Learning Journal, 3(2). 3*(2).

Stake, R. E. (1995). *The Art Of Case Study Research*: Sage Publications Inc.

Stake, R. E. (2000). Case Studies. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research* (pp. 435-454). Thousand Oaks, California: Sage Publications.

Stenhouse, L. (1975). *An Introduction to Curriculum Research and Development*

Stubbs, T. (2006). *Design Drawing in Instructional Design at Brigham Young University's Center for Instructional Design: A Case Study* Brigham Young University, Provo.

Tesch, R. (1990). *Qualitative Research: Analysis Types and Software Tools*. New York: The Falmer Press.

Tyler, R. W. (1949). *Basic Principles of Curriculum and Instruction*: University of Chicago Press.

Vidich, A. J., & Lyman, S. M. (2000). Qualitative Methods: Their History in Sociology and Anthropology. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research* (Second ed., pp. 37-84). Thousand Oaks, London, New Delhi: Sage Publications Inc.

White, G. L., & Sivitanides, M. P. (2009). A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers' Cognitive Characteristics. *Journal of Information Systems Education, 13*(1), 59-66.

Wiedenbecka, S., Ramalingama, V., Sarasammaa, S., & Corritore, C. L. (1999). A comparison of the comprehension of object-oriented and procedural programs by novice programmers. *Interacting with Computers, 11*(3), 255-282.

Wikipedia. (2010). Speech recognition. Retrieved July 2010, from Wikimedia:

Wiley, D. A. (2000). *Learning Object Design and Sequencing Theory.* Brigham Young University, Provo.

Williams, D. D. (2009). Educators as Inquirers: Using Naturalistic Inquiry Available from http://webpub.byu.net/ddw/qualitativebook/

Appendix A. Data Gathering Plan

Adapted from R. Stake guidelines for field observation case study (1995, p. 52)

1. Anticipation *(completed in prospectus)*
   - Case study literature
   - Identify the case
     - Case boundaries
   - Anticipate problems, attributes, persons, events etc.
   - IRB approval
   - Publication intentions
     - Dissertation, journal publications

2. Prospectus approval

3. Identify participants, Schedule visits
   - Contact institutions
   - Explain study outline
   - Recruit interviewees
   - Schedule visits

4. Further preparation
   - Identify Conditions and spaces for data gathering
   - Review instruments and standardized procedures as necessary
   - Specify/test recording, record keeping
     - Coding system
     - OK with host?
     - Back-up storage
   - Review and check priorities etc.
   - Reconsider issues or theoretical structure to guide data gathering
   - Sketch plans for final report and dissemination of findings

5. Initial visit
   - Meet with primary contact
   - Rules for observer and host?
     - Refine access rules if needed
   - Identify real or potential costs for hosts
   - Confidentiality rules
   - Review process for drafts to validate observations (member checks)
   - Revise plan if necessary

6. Gather Data, Validate Data
   - Make observations, do interviews, debrief informants,
   - Collect artifacts
   - Track inquiry arrangements
   - Select vignettes, special testimonies, illustrations
   - Classify raw data, begin interpretation
   - Redefine issues, case boundaries,
     - Renegotiate arrangements as necessary
   - Trustworthiness checks
     - Member Checks, audit trails etc.

7. Refine conceptualization
   - Identify possible "multiple realities"
   - Seek for emic perspective
   - Allocate attention to different viewpoints, conceptualizations

8. Follow-up
   - Identify areas where understanding is weak or thin
   - Re-contact participants as necessary
   - Acquire, analyze further artifacts as necessary

9. Analysis of Data
   - Review data under various possible interpretations
   - Search for patterns in data
     - whether or not indicated by the issues
   - Seek linkages between courses, structures, activities and outcomes and people,
   - Draw tentative conclusions, organize according to issues, organize final report

10. Provide Audience Opportunity for understanding
    - Describe setting in which activity occurred
    - Consider dissertation as a story, look for completeness
    - Draft dissertation and materials for audience use
    - Try them out on members of the audience
    - Help readers discern typicality and relevance as a base for generalization
    - Revise and disseminate

Reference

Stake, R. E. (1995). The Art Of Case Study Research: Sage Publications Inc.

Appendix B. Interview Protocol *(updated)*

Each respondent will be interviewed using the following protocol and question outline. This is a semi-structured interview in that it is guided by the questions listed below but the respondents are invited to develop the topic with their own words and thoughts. The term "respondent" or course designer" will be used to describe the interviewee. In most cases, if not all, this will be an IT instructor responsible for course design and change.

Introduction

1. Do personal introductions if necessary and indicate that participation is voluntary. Participants will not be identified by name in reports but materials and quotations from the interviews may be used. (emic perspective and qualitative methods).
    a. Explain use of recording equipment for the interview
    b. Briefly describe the open nature of the interview. Indicate that you may intersperse their descriptions with questions for clarification or to follow a line of thought.
    c. Obtain contact information: Name, preferred mailing address, email address and phone number
    d. Obtain formal consent for the interview. (signed IRB form)
2. Describe the research purposes. Indicate the desire to uncover the structure of educational design and how it is affected by changes in the design. Indicate the desire to reveal causes, mechanisms and effects of changes that may occur.
    a. Explain the concept of *member checks* and ask for their cooperation in reviewing and responding to summaries of the interviews.
    b. Describe the research questions in simple terms as follows:
        i. What changes when you change courses in IT? What inspired the changes and what effects do those changes have on the way courses are designed and how they affect the students?
        ii. How are changes made? What procedures, influences and methods are used and how do they affect your work as an instructor?
3. Identify the respondent's background
    a. Ask about their technical background, both academic and experiential.
    b. Seek to gain a sense of the participant's role in course design, their formal training in educational design and their experience creating and changing courses.

Identify and describe course changes

1. Ask about specific course creation and change experiences, Use leading questions such as, "Tell me about a course or course module that you have developed or changed?" or, "Have you participated in course changes that have been significant?"
2. Narrow in on a few specific experiences of course changes. Use questions that might lead to understanding the research questions and that might reveal the underlying structure of

the instructional design. As far as possible allow the respondents to describe their own experiences. Use questions only where there are holes in the narrative or unexplored areas relating to the research questions. Some sample questions are given below. The purpose(s) of the questions are shown in italics.

   a. "What was changed? How extensive were the changes? What aspects of the course were affected by the change?" *Identify nature and scope of change, Start revealing structures affected by the change. Look for types of changes in the answers. Were they changes in topics, strategy, equipment, procedure etc.*
   b. "Why did you decide to do this? What motivated it? What benefits were you hoping to achieve?" *Identify causes from the participant's viewpoint*
   c. "What documents, assignments or course descriptions did you generate as part of the change process?" *Identify artifacts that were created. Ask for copies if appropriate.*

3. Explore the change by discussing course documents, visiting labs, looking at equipment if these will add richness to the understanding of the change. Record these additional experiences with photo, video, audio etc. as necessary.
4. Repeat the above procedure (2&3) for a small number of course changes, probably two or three per participant. Attempt to get changes of different scale to broaden the scope of the inquiry. e.g. a complete course re-design versus changing a single lab assignment.
   a. Identify and describe a few changes without necessarily exploring all of them in depth.
5. Direct the conversation to system-level issues that may not have come up in the specific discussions. Identify methods, templates, project management, teams, funding, workloads and related issues.
   a. "How do you typically implement changes*?* How do you decide when to make changes?" *Identify the extent of standard approaches, templates, project management or similar meta-systems used. Identify personal attitudes and methods of the course designer.*
   b. "Tell me how the college/department supports or interacts with you on course changes? *Identify external influences, funding, obstacles, hierarchies, teaching assistants etc. Identify impact of external units such as instructional design departments (e.g. CTL at BYU).*
   c. "What was the student response? Was the change a success? What were the effects of the change?" *Allow the respondent to define 'success'. Look for effects of change on the student's learning experiences. What trade-offs were made?*
6. Be aware of changes in each of the layers the instructional design layers model. Be aware of layer effects that may be described during both the design and delivery stages. Consider layer-selection priority, decide which layer led to which aspects of the decision. *(See Helps & Gibbons article for discussion)*
   a. Content: New topics, concepts, equipment domain vs. metacognitive, desired response, framework vs. content etc.
      i. Cogn. appr. (heuristic & control strategies, learn-to-learn)
      ii. Bloom's
      iii. Systematic processes
   b. Strategy: More effective teaching. Different methods, learner vs. designer goals

        i. Goals, setting, social, scoping, trajectories, activity, timing, artifacts, culture, engagement, goals etc.
   c. Control: changes in student interactions, student communication channels, student choices in learning
        i. Intentions, language, conversational patterns, devices, fidelity, synchrony, software
        ii. Layers-related controls
   d. Message: Interactional conversations, channels of communication, implementation of strategy, etc.
        i. Design for local conversation, one to many representation, delivery-mode generation of representation
        ii. Interface to strategy layer
   e. Representation: new presentations, equipment, models, UIs etc.
        i. Charismatic
        ii. Only visible layer
   f. Data management: The "memory" of the design.
        i. New ways to organize and control material, grades, student records; indexing of material; sorting and organizing student responses, etc.
   g. Media-logic: New media controls, algorithms or ways for the control and message layers to interact between student, instructor and learning materials.

Conclusion and artifact collection

1. Ask the participant if there is anything more they would like to add to the discussion or if there are any questions they would like to ask about the research.
2. Identify specific artifacts discussed during the interview and ask for copies or access to them. Get copies and/or URLs. Ensure that electronic access to web sites is functional (*Are passwords or access permissions required?)*
3. Remind the participant that they will receive a summary of the discussion soon and be asked to comment on it or amplify it.
4. Make sure the participant has contact information for the researcher.
5. Thank them and leave

Post-Interview Follow-up

Within 48 hours of the live interview complete the following follow-up procedures

1. Check that records of the interview are valid. Audio, photo and video recordings must be copied and backed-up.
2. A rough summary of the proceedings must be made for audit-trail purposes. This summary includes
   a. Date, time, name(s), place and contact information. Topics discussed, courses identified. List of artifacts.
3. Initial analysis (See analysis protocol) and a member check summary of the interview must be created and submitted to the participant (electronically) for comment and expansion. This will include the following items
   a. Thank them for their participation.

   b. The member-check will describe the participants experiences and summarize the conclusions from their viewpoint but structured around the research questions. (See analysis protocol)
   c. Invite their comments, corrections and any further thoughts they may have.
   d. Indicate receipt and validity of artifacts (or ask for them again). ‘Validity’ in this context simply means that the artifacts are readable and relate to the topics under discussion.
4. There could be several cycles of discussion with the interviewee clarifying and expanding the discussion, based on the member-check document.
   a. Record all these interchanges.

Secondary follow-up

Within seven days of the live interview complete the following procedures.

1. Analyze the interview material. (See analysis protocol)
   a. Seek for structures, trends, methods and answers to questions raised in multiple layers of design architecture.
   b. Look for links in the material to the primary research questions.
   c. Are important new questions raised? Does the interview change the questions or suggest changes in the methodology?
2. Decide if further interview with this respondent is required
   a. Decide the objectives of the next interview. What are the unanswered questions or open issues? How can these be addressed? What kinds of information or artifact might answer these questions? How do these questions relate to the research questions?
   b. Decide the format, (electronic or face-to-face) in light of resources available (principally time and money).
   c. Schedule as necessary.
3. Write the audit-trail documents for progressive subjectivity check.
   a. Review the procedures used.
   b. Document any changes in methodology.

Appendix C. Data Analysis Protocol

The following steps provide a guidance framework for analyzing the data collected from interviews and other artifacts. Comments in italics indicate later modifications to the procedure.

#	Analysis Activity	Comments
	For each respondent do the following	
1	Gather data from interviews and transcribe	Include quotations and complete ideas from respondents' perspectives. It is not necessary to record every word but ensure that all possible interesting ideas and concepts are captured. (*later change: Every word was transcribed to capture nuances of meaning*)
2	Identify change events for course or curriculum	If necessary allow for duplicate and overlap single data points that refer to multiple events. E.g. if a single instructor comments once on multiple change events, include the comments in all such events. *(This was changed: The unit of analysis was modified to emphasize the process of change for each change, rather than focusing on the change event itself.*)
	For each event for a given respondent do the following	
3	Find key phrases and terms	Key phrases will emerge from repeated phrases and terms used by the respondents. Also use "eyeballing" (Ryan and Bernard, 2003) Key phrases are also defined by the research questions and the design layers discussions. (This was modified. *Key phrases must be interpreted in context to be meaningful. HyperRESEARCH software was used to search for and document key phrases.)*
4	Triangulate locally. Include information from artifacts and other data related to this event	Use documents, photos or other media acquired from respondent.

5	Identify potential themes using real examples	Include quotations, images, extracts from artifacts and keywords
6	Generate prototype theories of change. Link themes together in theoretical models. Iterate on steps 3-5 (and possibly 1, 2) as necessary	Allow narratives of change to emerge. Accept multiple viewpoints.
7a	Address the prototype theories generated to architecture questions. Refine theories if appropriate.	Can architectural changes be identified?
7b	Address the prototype theories generated to the research questions. Refine theories if appropriate.	
8a	Compare and contrast emerging theories	Refine the model.
8b	Use negative case analysis to further test the emerging theories.	Refine the model
8c	Identify what is not said: What are the underlying assumptions and values of the respondent?	
11	Repeat steps 3-8 for all events from this respondent	
12	Prepare a “member check” document and submit to respondent. Request respondent’s feedback	Use emic perspective. Summarize to minimize respondents cognitive load and to minimize demands on the respondent’s time. Respondents are not educational theorists they are computing professors and they are busy.
13	Address the theories to the research questions	
14	Create a “member check” document summarizing findings from the respondent’s perspective. Submit to respondent and request feedback	Be respectful of respondent’s time and viewpoint. Be succinct and do not overload the document with the research interpretations.
15	Modify findings based on respondent feedback	

16	Repeat steps 1-15 for all respondents
	For the complete set of data do the following
7	As a check, compare and contrast all theories across all respondents and globally test theories with negative case analysis.
18	Compile results into a comprehensive narrative addressing (a) the theories that arise from the interpretation, (b) the research questions, and (c) the architectural findings.

Note that this protocol was not strictly followed, but it provided a guideline to direct the research.

Appendix D. Identified Course Change Events

	Event	Interviewee (alias)	Comments
1	Changing Database course (4002-360). It also affected (created?) 4002-461	Susan	Restructured database intro course (360) because students struggled with theoretical emphasis. Put more "hands-on" earlier in the course and used Math classes to enhance theory understanding. Also created a database concentration of 3 courses (part of same change process)
2	Combined with changing database course (360) changed teaching strategy to an "active learning" model	Susan	Their "active learning" model is also called studio teaching by others.
3	Each year update database class to a new version of Oracle database software.	Susan	Annual updating to new version. Stay "one version back" rather than latest version.
4	Database Performance and Tuning class: Modified conventional course to an open discussion and independent learning approach with students	Susan	Change was not successful. Students did not like unstructured nature of class. Change was effectively reversed later.
5	Update "Programming for Digital Media" for new versions of ActionScript programming language (4002-434)	Jack	Multi-person/multi-year process. Changed from ActionScript version1 to 2 and later from v2 to v3.
6	First quarter with ActionScript class. Adopted matrix math to replace polar coordinate math.	Jack	Changed technical content to achieve better understanding. Change was not successful. Went back to simpler Cartesian/polar forms.
7	Modify class teaching and assignments to cater for students from two different majors. Attempt to meet needs of both groups (4002-330)	Geoffrey	Nominally the course has same outcomes and assignments for all students but they have different learning experiences. Graphics students learn coding, computer students learn graphics.

8	Updating ActionScript in New Media 230 & 231	Geoffrey	Learning to program using ActionScript. This change follows the change in the 434 course. (course dependency)
9	HCI course change	Lisa	Changing tools, changing teaching materials
10	2d Animation course	Lisa	Changed resource materials
11	Single frame animation	Lisa	Changed teaching methods because of student unhappiness with deadlines. Initial changes failed and then improvements achieved with a change of ID.
12	New material drawn into classes by student interest	Lisa	Class nominally remains the same (structure). New teaching materials added each tome. New technology introduced by students into projects and supported by instructor and lab staff. New technology "pulled" by students rather than "pushed" by instructor.
13	Adopted a "Call and response" teaching model. This is a variant on the "active learning" model her colleagues use.	Lisa	TA inspired the changes with spontaneous behavior involving students.
14	Rubrics for HCI (425 & 426)	Lisa	Designed new assessment rubrics. These affected the teaching of the class substantially. Re-focus of the class around the desired outcomes and assessments.
15	Redesign class websites	Lisa	Redesigned class websites to suit personal teaching style emphasizing graphics, de-emphasizing text. Class refocused on the main point of the class (search for "what's the point")
16	Introduce new material to IT104	David	Changed emphasis (added material) to a class at request of a teacher of a subsequent class in the sequence

17	Evolving web development class to include for new classes of technology.	David	Adding AJAX technology to labs for the course. Weeks of research on new tech. “Radical” re-design of labs. Teamed with TA. TA input changed the nature of the lab design.
18	Converted web development class to studio teaching model	David	Studio teaching vastly improved student acceptance of the class.
19	Changed technology to both teach in class and to support class (labs). Choice of virtualization technology	David	Improve flexibility and reliability of labs. Selected among options
20	Complete database course re-design to include new technical material.	Tom	Driven by desire to have a current topics class. Avoid problem of printed textbooks (too rigid and change too slowly)
21	Modification of Introductory Technology course. Coordinating class and labs. Example of digital lab	Tom	Had to reconstruct many materials and create new design for course due to weak handover from previous instructor and out-of-date technical materials. Used TA both for input and for lab design.
22	Creation of the Computer Storage and Peripherals class. New technologies added. Class is also regularly updated as new technologies appear.	Adam	Developed a new class to meet changing student and curriculum needs.
23	New textbook plus note-packet for Networking class (327).	Adam	Old textbook (2002) used (special edition reprint with extracts from chapters), plus augmented with additional material that the instructor adds each time the course is taught.

Appendix E. Member Checks: Summaries of Interviews

Summaries of the interviews with each of the participants are provided below. These were sent to the participants and where changes were suggested they have been incorporated into these versions of the summaries. Alias names are used in these summaries.

Interview: Susan: Summary of Interview for Audit (Member Check)

Introductory Remarks. Studying change in courses.

No formal background in educational design. Experience teaching deaf students at the community college level. Teaching software code and computer operation. Was a full-time teacher with NTID and became an unofficial TA for all the students studying computing, while officially serving deaf students.

Brief discussion of the development of the discrete Math sequence, and how it was tailored to the needs and interests of the IT students.

Course development: Database course. The IT students, "just hated the theory" but were very interested in configuring systems. The database class did not attract their interest initially – although some of them would realize its value later. Several related courses were developed to create a database concentration for the degree. The introductory course was designed by selecting a series of database related topics and activities. Math topics were moved to a Math department class.

The instructor adopted the "active learning" model popular within the department. Active learning is very similar to studio learning but they distinguish from the studio teaching model used by the college of visual arts.

> Active learning is really that opportunity to try to engage people on all levels, to try to hit all the modalities of learning, that people can *see* what you're talking about; they can *hear* what you're talking about; they can *try out* what you're talking about. So we're trying to hit them on all levels so that, when they're in the classroom they have a computer in front of them, and you'll be doing *something*. And so you roll through a section of content, talking about it, putting up examples, so they listen to you. You need at least half an hour at the end to give them a problem to work on, and then they have to turn that problem back in at the start of the next class.

Description of a course change: Change in a database class (4002-484) to make it more independent of the specific technologies they were teaching. This was done by focusing the design of the course on connectivity standards such as first the Open Database Connectivity (ODBC) standard and now JDBC, which are relatively independent of specific technologies and apply to many database structures. In doing this they also pursued a second goal. Students were asked to use three different programming approaches (languages) for working with data (ODBC, JDBC, .NET). The specific goal is to have students recognize the inherent similarities in these different approaches, to find a common pattern.

> We would implement that by giving out a bunch of assignments, and we'd talk about the concept. I'd draw this giant introductory diagram so you can see and talk about the various objects, each layer of that particular architecture, and they would have to go

> in and use the appropriate object in the code. I would give them a large program in which we had knocked out segments, and they had to fill in the blank spaces.

A large percentage of the course grade (most of the grade) would depend on a practicum test, where the students would have to demonstrate that they could successfully write computer code to accomplish specific database tasks.

As part of this on-going process the computer languages were changed, for example Visual Basic was changed to C#.

> We've always felt that IT is an academic discipline. We've always tried to stay on the side of "These are the basic IT concepts, principles, and this is what we can do. These other things are just our tools."

Despite this focus on concepts and principles rather than tools, E.G. "It isn't teaching MySQL, it's teaching database concepts illustrated by MySQL" they still need to change their technology frequently, as driven by the market. However in order to reveal the structure of the system to the learner they sometimes use less sophisticated tools (text-based source code (open-source)). More sophisticated tools (modern graphically-oriented drag-and-drop development environments) hide the complexity, but also hide the relationships.

Definition of success: "I think that change was very successful, and for me success would be that the students were motivated; that they feel like they got a lot out of the course, even if they don't love database." Multiple measures of success including student feedback, course assessments etc. On the other hand, they take formal measures because they are expected to, but do not actively use them for improvement. I.E. "And then we are supposed to measure that – we never do. Like typical faculty, we put the questions in and then we never evaluate the results"

One example of a less successful course change occurred with the introduction of a new database course. The instructor asked the students to work independently and then come back to class and discuss results. Unfortunately the students didn't work independently - they needed more guidance. The class was evolved over a couple of teaching cycles to a more traditional model (active learning model).

Managing change in the curriculum: "I don't think we manage it. I think it manages us. I think we are continually running ahead of the snowball." On the other hand, "There's always a new version of Oracle. I try to stay at least one version back, and not rush towards the new version, but there's a change every single year."

There are changes each time the class is taught:

> "Well, I have to run through every single exercise and I have to go through every single lab and rerun them, to make sure that instructions I've written are clear, that my goals are clear, that the assignment and software are still hitting my goals, or maybe there's a change in the way Oracle's doing things, and then I have to rethink the concept that I'm presenting, and has that changed, has the underlying practice changed? It's a big pain, but I essentially go through every single in-class, so each lecture has one to two in-class exercises, and I have to look at every single one of them and run it, every single lab and run it. I do those personally. It'll take me five to six hours to run through a lab, and then identify problems I would say, all told, it's probably eight hours." "And then you also have to deal with the in-class exercises"

There is some degree of team effort. The Graduate Assistant (GA) implements the new version on the lab machines after the instructor has developed and tested it.

The change process is continuous, "When I go though the class I always make notes on my syllabus. Okay, we could add this; we could add that, (etc.) and then I come back and I reevaluate the lecture section and say, okay, this is this new cool topic." "Last year it was I had to change every single thing. New image, new everything. Normally I don't change the lecture slides too much, except for the new concept"

Description of the change process: "I would say there's ongoing tension between the need to keep the curriculum up to date, particularly the computing curriculum. We always say, it's not Beowulf – we're still not beating Grendel with that arm – every year - it's the same thing. For us, every course is evaluated, and we set an acceptable level of change so that we would not have to dramatically change every course more than once a year. We would evaluate in the summer and decide what we're going to do, and try to keep that consistent across all faculty that teach the sections, and more or less consistent for the year Naturally you're going to have to change your labs, so that it's new for every group of people, so that you're not just handing in last quarter's labs. But the actual content shouldn't shift more than once a year unless something dramatic happens." Change discussions are done informally, up and down the hallway.

The effort of this continual updating of courses is significant, Susan further states that this is not specifically rewarded but expected, "Traditionally, that has been an expectation of the faculty, that the faculty will keep the curriculum up to date." In order to achieve this faculty are expected to work long hours. This instructor reports that working 70 to 75 hour work-weeks is common. But IT faculty keep on doing it because, "I think computing faculty want to be up to date. Computing instructors are strongly interested in computing. "

Course changes are largely a solo effort.

Course grades and materials are stored in the MyCourses content management system.

Interview: David: Summary of Interview for Audit (Member Check)

Introductory remarks. Studying change in courses, the research questions.

Most courses change all the time, they are dynamic. Some aspects change every semester. A few courses, such as IT104 are relatively static, but even that had a change recently to accommodate a need for a subsequent class. This was an example of a change due to feedback from the instructor of the subsequent course. It did not reflect a change in the subsequent class, rather that instructor thought the topic of transmission lines was being emphasized in IT104, but it wasn't. The change included adding a handout and some lecture time to IT104 to emphasize the topic.

The web development course is in a constant state of flux. Changing it is, "almost a labor of love for me, I put in an inordinate amount of work on it, constantly."

Suggestions for changes come from the IAB, from students and from the technical literature. However the class is "jam-packed full", so every addition requires eliminating something else from the class. The basis for decision for new content is how fundamental it is or how likely is it to have a long-term effect on the discipline.

The suggestion to add material on AJAX originally came from students. Their position working in industry sometimes allows them to discern trends that faculty in the "ivory tower" may miss. A few weeks of further research by MGB showed that this trend was indeed important and, after consulting with a colleague, the changes were made to the course. Google maps is a good example of the use of this technology, with its ability to interact with the user by right-clicking etc. but the impact of being able to interact with just part of a web-page is pervasive through the industry. About two years ago I noticed that, "Woah, hey! Interfaces are great!"

This technology became much more useful after improved standardization of Javascript and the Document Object Model, so that it would work more easily on all platforms.

Student feedback had indicated the need for adding AJAX to the course but had also indicated that they would like to see the labs modified to, "make something that is a better showcase of all our technologies".

One particular TA became involved in the upgrade process. He had strong technical skills in this area, was eager to become involved and did not want to do ordinary TA work. He also influenced the instructor to change the new design from stand-alone labs requiring different technologies, to having the students develop a game-playing website. Over the course of a semester an application would be progressively developed. The website design required the integration of multiple IT technologies (PHP, PERL, JSP, MySQL, etc.). The TA convinced the instructor that this would have higher student appeal. The complete new set of labs was created on a "just-in-time" basis through the semester, as the class was taught.

The relationship with the TA was collaborative, with the instructor in a team-lead role rather than in a "boss" role. – which is his preferred style with all his TAs. He also has gotten the Center for Teaching and Learning involved to help develop some animations for the website, but these have taken years to emerge, and are still not complete.

There were weekly meetings with the TA throughout the development period and the TA was developing and testing lab ideas during each week (working ~12hrs/week). There were about 8 new labs over the 14 week semester, some labs taking more than one week.

These changes were positively received by the students. The instructor uses two surveys to get feedback on the class, one at the middle of the semester ad one at the end. In addition to this there is a standard institutional survey for all classes taught at the university near the end of the semester. All the surveys indicated positive response from the students. They liked the idea of building an application that they can add to their portfolio as part of their resume'. The finished website design is used by the students to show current and future employers and family members.

> "Before the semester started we had an overall architecture sketched out, what we wanted to do where, through the labs of the semester. But then we slipped into the common, academic mode of just-in-time teaching and so we were meeting constantly arguing over the details of each one of these labs, and working to get that better. I was pushing him to make sure he implemented everything we were asking them to do ahead of time to make sure it was scoped properly. And then we would take his stuff and we would basically record it showing [the students] on screen the demonstration program of what they needed to have done at that point. So it was a busy semester."

Much of the original concepts and even technology was retained, although organization and emphasis changed significantly. Some was reduced (eg PERL), which created space in a very full curriculum to add new material (EG AJAX). The processes were also changed to align better with professional practice expectations (best practices).

Another major change happened in parallel with these changes to the labs. The class was re-structured around a studio-teaching model. The instructor is in the lab with the students for all class interactions and the class time is mingled between instruction and student exploration of the topics on the lab computers.

These major changes all happened within a single semester, about two years ago. Since then there has been a process of consolidation. For example, "[The TA] is a prolific IT person in a lot of ways, but … his lab-assignment write ups needed a lot of clean-up." The overall structure of the newly designed course has remained. One additional development has been the provision of resources to help the students understand the concepts. "

> This course, to a large degree is Internet material based. Imagine this: the best resource on how to program the web is the web. Better than books, or anything else. So we've been finding the best of the best and going through it and making it accessible to them. (MGB, ~xxx)

Another interesting factor in this development is that a cadre of TAs that are competent in the subject area has been generated from those who have taken the class. These TAs improve the class for the students and also improve the class ratings for the instructor.

There have been several indicators of how popular the class has become with the students. One is that the students cite the class as one of their favorite classes in exit interviews. Another measure of success is that Computer Science students are choosing to take the class as an elective class.

Since the initial set of changes a number of other changes have been implemented. One change was the institution of taking attendance. The instructor was personally opposed to taking attendance as a philosophical viewpoint. He didn't want to seem to be, "beating them up over it." To make attendance a less intrusive process an attendance server was created so that the students just need to log in when they arrive in class and that serves as their attendance registration.

Since making these changes the class has been taught using the studio model twice. The first time was part of the transition, "it was pretty shaky," the second time the improvement was marked (15-20%).

Another interesting facet of the course is that students are encouraged to share their findings, “Fix it, and make it look good. And when you find something, yell it out to the rest of the class so everyone else can find it too.” (MGB)

Despite the open studio model and project orientation of the class, it is quite clearly structured. Sometimes a student with web experience will propose doing his own project rather than the class one but the instructor “gently discourages it”

The studio model of teaching encourages collaborative learning and peer teaching. “I'm depending on students teaching each other in the course. I encourage it, I almost threaten them if they don't do it.” This is closer to their authentic future professional work environment. “I think that a realistic way that this type of thing is done in this industry is group dynamics. You work with other engineers you don't work on your own.” To further encourage this mode of learning there are no penalties for sharing, “You [can] all get A’s.” There are mechanisms in place to prevent plagiarism.

There were unexpected benefits of the studio lab approach. One of the early driving motivations was to help students get more involved with the labs, to get started on them well, and to thus reduce the failure rate. Having started using them, other benefits are coming to light, such as shared learning and group encouragement. Prior to the studio approach good students would take 5-10 hours to complete the lab but weaker students would take 15-20. Most of the students manage to finish the labs in about the same time as the good students. There also seems to have been a spillover of collaboration between students. There appears to be more general collaboration outside the assigned studio-lab times.

The instructor has an on-going internal debate about whether to add new technologies to the course (EG Python). Sources for suggestions for change include students, technical press, IAB, on-line blogs (eg Slashdot). Sometimes changes are forced upon the class. For example, the class uses virtualization technology. Previously they were using VMware, but the company changed it and “broke” it, so now the class has switched to Virtual Box.

Faculty and students welcome change.

> “I tell students if they don't like change, they are in the wrong major, they need to go into Geology, where change is much more glacial. Hey, I chose this career way back when, I think because I kind of enjoy being on the leading edge.”
>
> “And do you think the students enjoy it?”
>
> “They're in the wrong major if they don't. I really believe that.”

The Blackboard course management software package is used to manage data for the class. Students submit assignments through Blackboard and grades are posted there.

Constantly changing the course takes a lot of time. The instructor says he can, “easily spend two or three times more time on this class than I do on the others.” He has also incorporated the class changes into his scholarly work by discussing changes in teaching in some venues, but feels that it has not been well accepted as scholarly work by the university administration.

Teaching is focused on concepts, which can be represented very effectively through technology.

> I'm really concerned that they get underlying principles of the way computers work, as far as how to administer them, how to program them, how the state machine model pervades computing and some people really struggle with that concept of sequential fetches and steps, but those people, if you don't get the basis of how computers do these types of sequences, they'll never be good IT people. So, for me, success is anything that builds towards them really getting the underlying how things work in

computers, and being able to apply that. That's why I said it was exciting to me when I discovered AJAX was based on asynchronous calls. That's an underlying concept that they weren't getting anywhere else. And AJAX 10 years down the line will probably be dead and gone, most likely. But likely will there be some other technology that has a similar foundation. And they are going to understand they can have these different servers over here working independently, and they're going to communicate with each other through asynchronous calls! This is a good concept for them to know and understand.

End of Interview

Interview: Tom: Summary of Interview for Audit (Member Check)

Introductory Remarks. Studying change in courses, the research questions.

Some formal background in education – the instructor took a required overview ("shotgun") course as part of his PhD.

Changing courses is common, "It seems that about every course that I've ever taught, I've had to come in and change it." He also had to change courses because of technology changes, "IT changes so much, that I think that was another thing unique to our discipline, and I think that is a key fundamental change that you don't get in Art or History, or frankly, even over in IS, you've got to keep up to date on the different tools, but that's different from the actual technology applications." And also commenting on the speed and nature of changes in IT education, "IT changes so fast and updates so much that, it's one thing to be able to use it, it's another thing to be able to design it, or integrate it or apply it to an organization, and to do that you've got to stay up on top of things, which, on a side note, is why a lot of my classes are getting rid of textbooks."

One of the key attributes of IT education is that the students need to learn to integrate diverse computing environments and to "fit technology to solve a problem in the organization, or for a person, or for a family."

Two course change experiences were discussed. There were several motivations for these changes. The first motivation for both changes was that the instructor was requested by the program chair to revise and update the courses. The next motivation was to bring the class more up to date technically, relative to current professional practice. The next motivation was to convert the class to more of a "coaching approach" rather than a teaching or "instructional" approach. This approach is one favored by the instructor based on his own experience and background.

The first example of a course change was redesigning the Database class (IT 350). This class had previously been taught by two different instructors and seemed rather disorganized. The instructor redesigned it with a new class structure and new textbook(s). The textbook is an "open-source" textbook, consisting of online materials that can be altered and updated as necessary.

The course is designed as a "reverse onion" by building layers rather than peeling them. He describes the process as follows:

> "You usually talk about peeling off the layers, well we're building layers; we start with the core and we build. And the core is, we give them a little practice then we give them the theory behind it and giving them more theory and it's expanding out and it slowly grows out, so by the end of the course they can do anything, one to the other."

Another example of a course change was improving the IT101 course. There were some problems with the class prior to it being re-organized. One problem was a lack of continuity or sequencing. One change to improve this was to give students a "boxtop to the jigsaw puzzle so they could see how all the pieces fit together". Success in learning is achieved when the students struggle to complete an assignment and then 90% of the students complete the assignment successfully.

Amongst other issues the labs needed to be coordinated with the lectures more. Several measures were taken. One is demonstrating parts of the lab to the class (Digital Principles Lab).

Several exams and the class project were changed or eliminated based on feedback from students and evaluation of student workload.

Labs are intended to combine skills, abilities and knowledge. A lab assignment might be to build an Ethernet cable, but students will develop knowledge of networking and problem solving in the process.

Significant goals for future improvements in the 101 course include better organization of the learning experience for the students and better documentation of the course design, so that it could be shared with future instructors of the course.

This redesign of courses consumes considerable effort. Currently the instructor estimates that he spends 75% of his time on teaching (including updating courses), 15% on research and 10% on service/administration.

Notable theoretical influences on design of courses are constructivism, Bloom's taxonomy, an approach described as designing a course based on student outcomes that include "Knowledge, Skills and Abilities" from an Training Systems Design course in his PhD (e.g., "Evaluating Training Programs" by Donald Kirkpatrick, "Train-the-Trainer" by Itner & Douds, etc.) and Dick and Carey's "Systematic Design of Instruction". In addition "McKeachies Teaching Tips" also provides input.

The course design and changes are done as a solo effort, with some input from other instructors and other sources. TAs have also provided helpful input and in some cases modified parts of labs, under direction of the instructor.

All courses are administered through Blackboard CMS. Only one class has a separate course website.

End of Interview

Interview: Geoffrey: Summary of Interview for Audit (Member Check)

Background:

Promoted collaborative work through the Center for Digital media. “I was very interested in collaboration. So one of the things that I did very early on is I got collaboration between IT and the people in the art school”).

Helped create the Center for Digital Media. Created a degree in New Media. (Three degrees, A BS in New Media, New Media Interactive Development, and a BFA New Media: Imaging and Design. Teamwork projects and industry involvement.

His characteristic teaching mode is that of “Active Learning.”

Description of “Active Learning”: Teaching takes place in a lab, with the instructor’s computer connected to a projector. Start with an example (eg a piece of code), share the need for what needs to be done (specifications) and sample code with the students, active discussion with the students as it’s being presented (and developed?), and then at some point hand the problem over to the students and specify how it is to be tested. Students then develop on their own and either finish it in class or after class. While the students are developing the instructor and TA will circulate in the class and interact with individual students to try ensure they are all moving towards successfully completing the assignment.

The “Programming for Digital Media class:

Description of the dual nature of the 434 class with two different audiences, one from the New Media students (College of Imaging and Arts) and the others from the Gaming degree program (Computing College). Designers (New Media students) may be interested in applications of the ideas while Gaming students may be interested in adding features or solving technical problems related to the course work. All of these students are building interactive applications in 2D (bitmap-based) and in 3D.

Mixed IT and design (art) – creates problems for changing courses: EG 4080-4342 Required course

“I’m questioning now: should I split it? Are those two audiences really looking for different outcomes from that course, and for that matter, bringing different skill sets into that course? And do I have to really separate it, even though the gist of it is the same?”

“… be a set of requirements that are fundamental to the outcomes of the course, and I said, well, “Your solution to this problem or your application you make from me must have these things to get a grade of B or better.” But to get an A, I find myself making a list where people have choices, and saying, “Well, if you’re a designer, you might want to show me an interesting application of this, but if you’re a programmer, if you’re coming out of the Gaming group, which is much more technical, I’d like to see you pick one of these problems and solve it, or add one of these features to the work we’ve got so far,” and so I’m looking at, in a sense, two different inputs and two different outputs coming in the same course”

---

2. Course 4080-434 Programming for Digital Media: Course Description: In this course, students will create object-oriented interactive applications in domains such as simulation, gaming, instruction and artificial life. They will build data structures, and classes to create virtual worlds in 2 and 3 dimensions, populated by autonomous agents. Programs will often extend modules created by previous classes or the instructors. Some projects may require working in groups.

Two sets of objectives: The first objective is to use tools and concepts for building graphic worlds and the second objective is to complete a project to design a world.

Different outcomes for two sets of students in the same course. One set gains 2D and 3d world development experience, which leads directly into their professional careers, while the Game Development students see the same experiences as an overview of the kinds of interaction possible, that will alter be implemented with detailed interaction with hardware and high-performance languages (C++)

Theoretical approach TEACHING STRATEGY:

Interacting with students to determine levels of learning and adapt the instruction to meet needs. “I'm writing some code and then I'm saying, so what do you think we're going to do next? I'm looking for somebody to suggest it now” (Geoffrey, 26:15). They don’t “get it” so the active teaching strategy breaks them up into small groups and then uses specific questions for each group to explore the possibilities. This actively engages the students in the learning process. Also use a random selection strategy (pack of cards) to choose which students should respond to a question. This requires all students to participate and to be ready to respond. This works much better than the previous strategy, which was just asking students to respond.- that strategy tended to lead to, “getting into a fun dialog with my three or four best students.”

“Active learning” where instructor leads discussion by introducing a topic or concept, students explore the concept – using computers in the classroom – while the instructor circulates and interacts with them.

The class has changed multiple times. “I'm almost forced to change it to some extent by small things” Originally Flash Script was a menu driven language but, “Eventually it became a language that supported an editor and functions and parameters that you could teach from, but recently it's become a very powerful object oriented language that I can really express things with, and I now for my media, in other courses, we use that as our introductory programming language, instead of JAVA or C sharp.” The ActionScript language also became a rich object-oriented language and is used as an introductory programming language , in preference to Java or C#.

This rapid evolution of the language creates a secondary problem. “Our worst case scenario is the student who comes in, takes Programming I, gets a C, doesn't have a great time, shies away from Programming II, finds they're a senior and they owe it to us, and goes back to take it, and not only have they forgotten everything they learned, but the language has taken two quantum jumps since the one – it's very difficult.”

When these situations occur, “it’s a major headache” and so significant efforts are taken to avoid the situation arising. If they do happen the students are invited to sit in Programming I a second time (without charge) and then take Programming II because it is every difficult to take Programming II wit out an understanding of the [current version of] Programming I.

In other words,, although the title and the nominal contents of Programming I have remained somewhat constant, it has, in fact evolved into a different course – as has Programming II. The course is still based on the original outcomes but requires new pre-requisite skills. If students are not up-to-date on heir skills they will be at a disadvantage – even though, by strict curriculum guidelines they have the necessary skills on paper. Despite these changes the formal outcomes of the course, which focus on fundamental programming concepts, have not been changed. The outcomes are written to be language-independent, as far as possible.

wrt Programming 1 & 2 courses3 and the 330 course. These are alternative pre-requisite

Major changes were made in the summer. First the change group (three instructors) studied the new language (Actionscript), "we all got together and for, every day for about three weeks, we just went through that documentation"

The change to object–oriented programming (OOP) happened two years ago (using ActionScript 3). They needed to gain the support of the College of Imaging Arts, whose students would also be affected by the change. Convincing other departments:

"… we had a talk to the people in the College of Imaging Arts, and said we're going object oriented, using Action Script 3. And two of the folks over there said oh great, and the third one said designers don't do that! We're gonna have a problem. But when we got their interest up they went with us, okay. We did bring the designers along into Action Script with us. "

In order to teach the ideas the sought "sweetheart applications" that encapsulate key concepts of OOP. Possibilities exist for these in multimedia programming. For example, consider a character as an object, "He's an instance of a class … because he has this data about him. His X position, his Y position, his rotation, his scale, and he has these methods. He can turn left, he can turn right, he can go forward, … and so on. … So here's a class that describes these attributes and these methods" this provides, "A concrete way of expressing an abstraction that is easy for people to get their head around, and then eventually we can move to a more abstract notion of that. But we find that it helps those students a lot to have this very visual, very concrete way of looking at those concepts."

Fundamental concepts of programming remain substantially constant as the course evolves, such as functions, parameters, repetition (looping), conditional statements and so forth. These, and similar concepts, would need to exist in any introductory programming course, however substantial portions of the class do change, such as the introduction of object-oriented programming. There is a deliberate attempt to write the class objectives so that it looks software and language independent but, "I know it's not true." Although simple concepts such as programming constructs remain constant the changes in the use of different languages (such as Actionscript) and types of languages (object-oriented) effectively mean that the course changes substantially and needs to be revised on an on-going basis.

In addition to the content changes the strategic approach to teaching the class has also changed, with the adoption of the active teaching model.

A good instance of the changes in content is illustrated by the assignment teaching Turtle Graphics. This was previously done in ActionScript 2 and now is done in ActionScript 3 (object oriented).

REQUEST A COPY OF THESE BEFORE-AND-AFTER ASSIGNMENTS.

These changes were also influenced by feedback from placement data. The best students were taking additional CS classes, which got employers "pretty excited" As a result the programming course sequence will be extended further. It would be desirable to produce "Leonardo" type graduates who are professionally skilled in both technology and design and the New Media program turns out as many of those as we can. Large companies will still hire separate design and technology specialists but the students should all be at least conversant with both fields.

---

3. Course 4002-230 Introduction to Programming for New Media and Media 4002-231 Programming II for New Media. *Also* 4002-330 Interactive Digital Media

Course design has a focus on activities, "when I thought we'd develop the next version of the course my focus is usually on, 'What are the students going to do to learn this?' " as opposed to lists of topics."

Dual goals in instructional design are: "I believe if I can have a student create his own cognitive structure for something, that incorporates the concepts I'm trying to teach, that there'll be two benefits from it. … the two benefits that I'd be looking for was superior retention and a greater ability to apply this in a new context, which are two very important goals for me"

Discussion about creating a simulation to improve learning, which became a SIGGRAPH paper. "[We] realized that some of the things that people were doing simulations about were rather complex, and that students were doing a lot of research and understanding it in a very, very different way than the way they understand something from just reading notes."

The institution's model for reward is changing and, while curriculum and professional development are valued, the reward system is emphasizing research and, especially, receiving grants. In order to adapt to this some educational development efforts have been modified to include not only curriculum improvement but also opportunities for scholarly work about that development. IE the curriculum development is modified or extended to included measurements for research.

IT changes constantly. "I feel like I'm in a discipline that changes and has changed so much in the time that I've known it, that it's just a way of life." "I feel … that we are evolving over time, that our outcomes are changing as people are being prepared to enter a different world. Our inputs are changing, … the students have coming in to RIT and in fact, in particular with our students who are so digital and computer savvy, their way of communicating and learning is changing."

Graduate and undergraduate students are used to help in the development effort.

Assessment: Need to use quizzes and practicum exams (coding against time) to provide a measurable assessment.

STRATEGY and PRESENTATION LAYER:

"… we spend a lot of time in our particular way of doing things, trying to find these little sweetheart applications that … in multimedia it's easier to do in some areas, where you can find some application you can build, that is almost like a description of the concept"

Example: "… If I say, here's a character that moves around on my screen. I could say that that character is an object. He's an instance of a class. Why? Well, because he has this data about him. His X position, his Y position, his rotation, his scale, and he has these methods. He can turn left, he can turn right, he can go forward, he can – and so on. And see, so here's a class that describes these attributes and these methods"

"A concrete way of expressing an abstraction that is easy for people to get their head around, and then eventually we can move to a more abstract notion of that. But we find that it helps those students a lot to have this very visual, very concrete way of looking at those concepts, and then we say, okay, all of these characters need to do those things, but then some special characters are going to have other methods"

" … you spend a lot of time in the course design trying to find those nice little apps that have a very nice fit with the concepts that we feel they need to have as outcomes – and I's not so hard to do that."

Comparison between CS and imaging media department strategies:

"What the Java people do to develop their courses I see that they usually wind up with a stack of PowerPoints, and everybody goes into class with the same stack of PowerPoints. For some reason none of us do that. What we would tend to do is we'd get the outcomes, and from that we'd get the projects, and then we'd do things called ICE, In Class Exercises that support that active learning I was talking about.

Design approach is outcome-oriented:

"If it was something that I felt I had to talk about a lot, as opposed to code about I typically would, … come up with … some questions or talking points, [and] say, here's four questions that we'd like to be able to answer at the end of this talk. We might even ask people if they can answer them right away."

Summary of Course Design methodology:

"the set of things that we would develop for the evolving curriculum would be these projects, these in-class exercises, talking point questions, and then quizzes, practicum exams, and then it's a course."

The basic outcomes – measured in terms of programming concepts – did not change. New concepts did get added, EG object-oriented programming.

Design for change: - Hopes and reality

R: "Do you deliberately try and write it in that way? So that it doesn't change?..."
G: "I try and write it so that it looks language independent, so that it looks software independent, but I know it's not true.")

"I believe that people who understand technology are going to be among the most creative designers, because I think they have insights and how data is represented, to what is possible and to how it's done, that often lead to greater utility in the metaphors that they create."

Motivation for change:

"When all the New Media people sat down, those of us who were passionate about programming looked at the new language and we were very excited. We said, we can do things correctly now; we can do things better."

"This is – I'm sure you've talked to the chair about this – one of the most diverse groups of faculty that I've ever seen in a department. We've a couple of library science folks; we've three or four MFA people; the fellow next door has his PhD in Psychology; we've a mathematician or two; we have some people who in their hearts are really musicians; it's a very diverse group. Most of them, I think, see computing as a tool to do something, other than, wider than computing, they don't see it, study it only for its own sake."

Geoffrey educational background: Taught math in school (3rd grade through HS). Also "I kept getting hired to teach calculus"

Approach to instructional design

"There are still a few things that they taught me that I would say made me develop curriculum the way I do. … I was working for the board of education, it was required that I have a written lesson plan … ? And there would be something there. ??01:21:32. And that in that there'd be things like objectives, [but] I don't see it in college curriculum, was the … conscious action would be activities."

Focus for design is on learning activities, rather than a list of topics. QUESTIONS HE ASKS FOR DESIGN: "what are the students going to *do* to learn this?"

Claims constructivism as conscious choice of approach. Description of his version of constructivism, "if I can have a student create his own cognitive structure for something, that incorporates the concepts I'm trying to teach, that there'll be two benefits from it. … the two benefits that I'd be looking for was superior retention and a great ability to apply this in a new context, …"

Use of analogy to teach abstraction:
"a lot of what it takes to apply a concept in a different context [are] analogies." " If they can … see analogy, they can see where something which … deep down has some of the same concepts. Then I think they're on the way to being creative problem solvers"

"To what extent does the institution, your department and RIT as a whole, acknowledge and reward what you're doing?"
At the department level there is a recognition of the value of redesigning the technical content of the curriculum but, increasingly, at an institutional level there is pressure to increase traditional scholarly output, and particularly grant writing for funded research. There is a need to include curriculum redesign into scholarly work and publishable research. This is seen as an imposition, or at least a s a difficult thing to do,. It must be balanced with a teaching load of seven courses a year and the rewarded expectations of the institutions.

This is a difficult expectation in some regards, "I did a Masters in Computer Science and I did an MFA, and neither one taught me how to do research at what I consider to be the appropriate level to be publishing in the field."

Learning by having students create simulations.

Research is into education of technology rather than new technology.

Re-design team> "Geoffrey A___P___ (colleague professor), undergrad TA's (built some of the demonstration/lab projects). There are facilities on campus (a faculty teaching and learning institute). Does not use them. Has personal experience having helped other professors to develop courses while a (grad) student and now uses own skills to develop materials.

Example of something that didn't work. New C# course for New Media students. Students were not well prepared and course was not successful. In future they will do more work on prerequisites and on adapting the course to the background of the students.

"I feel like I'm in a discipline that changes and has changed so much in the time that I've known it, that it's just a way of life."

"I feel like it has to be that we are evolving over time, that our outcomes are changing as people are being prepared to enter a different world. Our inputs are changing, what the students have coming into [the university]."

End of Interview

Interview: Lisa: Summary of Interview for Audit (Member Check)

Introductory remarks. Studying change in courses, the research questions.

Introduction and Background in Sign Language and interpretation provided opportunities for observing many professors' teaching styles and helped her develop her own style. She observed common characteristics among all the effective teachers. Comment, "the effective teachers did the same things, regardless of discipline, which I thought was really, really interesting." These characteristics included structure – even in arts creativity flourishes better with constraints.

The underlying technology of IT courses changes, "a lot" unlike some humanities or other disciplines. "I'm still having to relearn things." In the past she focused more on learning the tools but the focus has shifted now. Major changes in her curriculum are more order and concepts. The tools (technology) should suit the creative needs, not drive it. Students are encouraged to adopt this attitude as well. She used to be concerned that students knew some technologies better than she did, but is less concerned since she realized her instructional value lies in her "eye", IE her ability to recognize the development process and "be the muse" for her students. What she is teaching is not the tool but the thinking process necessary to create.

Nevertheless, the institution provides current tools and the instructors and students use. Being aware of new developments in technology is done through websites (e.g. Lynda.com) amongst other means. Industry standards also have strong influence on which technology tools are used for instruction. Industry standards do not evolve as fast as they used to and thus it is possible to keep the lab updated.

Lab provides the students with a number of current industry standard tools but students are welcome to find and use alternative tools, if they wish. Students are encouraged to focus on their creation rather than the tools. Using various tools is often taught by individual instructions on an as-needed basis.

There is no standard "textbook" for the 2D animation course. Class instruction covers concepts and ideas. ActionScript is provided as a tool. The instructor uses "Tumblr" (http://www.tumblr.com/) to share material with the students. "Tumblr" allows the user to "bookmark" websites that you wish to recommend to others. Students actively contribute materials by making their own recommendations, which the instructor then adds to the bookmarked list. Communication with the students uses email and instant messaging (IM).

In addition to input from students other sources provide input to enrich and change the class. Some of these sources include colleagues and blogs (RSS feeds through netvibes). Enjoys "consuming media," watches video-games but does not play, does not enjoy first-person – shooters (the punchline is always "and they're dead".).

Change in courses focuses on "emphasis and order". I.E. Course structure and strategy rather than implementation technologies. Once again technology must serve creativity and design skill, not lead it.

Two different approaches to instructional design were discussed. Both are used in most courses. In all courses the concept of "call and response" is used.

> I will go and lecture and demonstrate, and then they stay in the room and work. I require that they stay and work with their peers, because it really is a studio class, they get to feed off each other. If they leave early, they're missing out on getting harassed by their peers, and also seeing what's going on. I discovered that it's actually a good idea,

while they're working, to keep playing animations, so instead of having music, they're, they multitask, which is something I cannot do. I find it fascinating that they work while music is playing, but more so they can work while the television is on.

This "call and response" strategy is clearly structured along studio-class lines. The instructor provides an assignment and then visits with each student as they develop their work. On a parallel level the TA, or other students, or the instructor, are sharing ideas with the whole class. The instructor will interact with students individually to either challenge them to explain the structure of their work or to coach them in specific aspects of it. This technique is used in all classes, both undergrad and graduate.

The other approach to instructional design is that instruction is a "story" with a story's structure. It is best illustrated by this quote:

> I think my philosophy regarding teaching courses is that it's a story that I'm telling, over the semester or quarter. And the story has a theme or an ending that I want them to get by the end, and every chapter should a beginning, middle, and end, and should support the theme. So the moral of the story of HCI-2, is "don't make me think," if I can put it down into a sentence.

Grading is done with a rubric. The rubric was originally intended as a guideline. Some students chose to interpret the rubric as a precise set of directions for the assignments and did sub-standard work designed to "game" the rubric. The rubric was then adjusted to take these issues into account. The rubric is focused on the product rather than on professionalism expectations. Uses a student self-assessment rubric that invites students to evaluate their effort in comparison to other students in their team. It has proved to be very effective in identifying the "slackers" in teams. Changes to the rubric solved the problem, but she constantly "tweaks" the rubric to improve it. This tweaking is based on on-going learning from student submissions.

The rubric avoids having to specify everything by providing examples of previous work. This circumvents questions about "How long does it need to be, how big are the margins?" etc. Students are also capable of recognizing bad work in other students' submissions. If they are shown a selection of papers with various grades then they can identify the differences without having to spell it out.

In their creative work students are encouraged to look at other students' work and feed off of each other's work. This is not copying because students want to be unique and are somewhat competitive, in the sense that they want to create something uniquely their own.

New material, particularly images, is constantly added to the class. This necessitates also removing some of the existing material. New material is sometimes selected because it fits in currently interesting themes. Old material that does not fit into these themes is removed. Newly developed technology, such as the iPhone interface, may drive a theme and thus dictate what is kept and what is removed. Aesthetics plays a significant role here.

She uses artifacts (EG Bopit) to start class discussions. The students have some background from previous classes in HCI to enable them to participate in the class discussions. (EG what is affordance?).

In modifying the 4020-426 course she converted the class materials from having "a lot of words" to new materials with almost no words ("weezified"). On first becoming one of the HCI professors she changed the discussion of the HCI-2 curriculum from coverage of material to, "what's the point?"

> "HCI is a process, that part of the process is usability testing, that is not a compelling story. HCI-1 is primarily about affordances and how people interact with

things. HCI-2 is a focus on - that you are not the center of the universe, and someone else doesn't think like you necessarily, and it takes two to communicate, not just one. Process and usability testing wasn't the exciting part for me, the exciting part was the story of communicating with somebody and trying to make that effortless."

Course development is generally done as a solo effort although sometimes she consults with other professors and instructors. Grad students are not used in the development process, neither is any RIT group used in helping to develop websites or other instructional material. Faculty peers are supportive. When there are multiple sections of a class the instructors of the different sections trade material and ideas but do not generally try to impose their version of the class on others.

In general the institution from the department level upwards is does not take significant note of course change efforts, they are considered part of the job and are not specifically identified or rewarded. Course objectives and course outcomes are controlled at a department level.

"My Courses" is used to manage grades and data. Websites are used for course materials.

Changes:

Code changes all the time (EG Adobe scripting). "Thankfully, I've been moving away from teaching any kind of coding, so the applications aren't changing as much. The things that I change in my curriculum now are more order and concepts and not, and I'm less concerned with technology"

"I don't learn the tool, I learn to figure out which tool I need based upon what I want to create."

"I used to be more concerned that I had to know more than the students did when I first started teaching, and I'm less concerned about that now. Because what I have that they don't have is the eye, and understanding of process. And the technology changes so much, I find often times they will come up with an idea that I may not have the technical ability to do, which is an excellent exercise"

"because what I'm teaching ultimately isn't the tool, it's the thinking."

She is driven by Industry standards for selecting software tools for labs. It is not changing very fast any more. It has "shaken itself out." Budget and convenience considerations often drive selection of a lab tool. There is a belief that the tools can all teach the same effective concepts.

There is an interaction between life and learning. Students learn outside the classroom "… when basketball players now do a brilliantly good move, they're like "put it in the game!" Because they have their own avatars there."

The content does evolve and that evolution is impacted by several influences. Sometimes the evolution is driven by changes in the audience rather than changes in the technology. Yet the technology maturing also influences this. Changing technology permits more realism, which may be accepted or rejected in the quest for better quality of product. Avoiding a preoccupation with the realism, IE the technology, and rather pursuing the essential, IE the story.

Some comments about teaching strategies:

"I discovered that it's actually a good idea, while they're working, to keep playing animations, so instead of having music, they're, they multitask, which is something I cannot do."

I do laps and look at every single person. When they're coming up with their projects I sit with them individually and say, "what are you doing? What do you plan to do? You're supposed to have thought about this by now. You better have an idea by the time I come back around my next lap, and you're definitely not leaving until you have an idea. Then I'll go around."

"OK, I want to do this," and then I'll go "ah," and then I go to the front of the class and say "Matt wants to do this particular thing, here's how you do it." And then I'll show a demo to the entire class, and that happens often. So whether or not it's in the curriculum, it might be, it's like okay"

"… being very responsive to the students is pretty much the crux of it"

How to choose new material:

R: "you'll find a theme, that that theme is driven by what you feel you want to do."
EO: "It's purely what makes a better show. "

One example of "weezifying" was the class PowerPoint slides migrating from lots of words, less graphics to lots of graphics and few (no) words. See files (01-2003-1.ppt (wordy) vs. 04a_AdvDesign-1.ppt (arty) – also see discussion later about tiny text in graphic slides). This is an example of course material that was changed from an earlier instructor's version to her own version.

There is no formal mechanism of control for changing classes. The instructor (class coordinator) owns the class and can change it fairly freely. Course changing has very little recognition as a formal faculty activity. "They're not very aware of the changes we're making, unless we make changes in the objectives and outcomes. We have a lot of leeway on how we teach the content. Note that Course Objectives and Outcomes are controlled at a departmental level. Course Objectives and outcomes are considered the item with the least degrees of freedom (most constrained) when redesigning courses.

End of Interview (Lisa)

Interview: Jack: Summary of Interview for Audit (Member Check)

Introductory Remarks. Studying change in courses.

Education: PhD in CS and engineering . His dissertation relates to e-learning, but he does not claim a formal background in education.

Research interests (from web page) “game engine design and development, entertainment technology systems, game engine development for social/cultural/educational/artistic uses, computing education, desktop virtual reality, computer mediated communication, computer supported cooperative work, computer supported collaborative learning, web technology design and implementation, operating systems design and implementation, distributed systems, behavior-based robotics and agents, architectural considerations for human computer interaction, and embedded systems”

He has rewritten, modified or evolved about seventeen courses in the last five years. “We've always had that problem in IT, which is with expansive growth in different directions that people can go into, there's a lot of pressure on curriculum and program change as well and really having to do the homework to keep up to date.”

Success is measured in different ways. It means that students can succeed in subsequent courses, but more importantly it the class is a relevant foundation for their professional practice (or graduate work) after they graduate. In addition it provides them with, “the level of flexibility that they can keep adapting once they're in industry.”

Description of a change: adoption of object-oriented ActionScript into the curriculum. “We'd just switched from Flash and ActionScript that was non-object-oriented, and were just starting to get into the object-oriented paradigm for how to program in ActionScript. And it was a fairly fundamental shift”

Changes to the course were driven by this shift in technology (newer version of the language) but the changes affected many other aspects of the course. The changes were at least partially mandated by their licensing agreement with Adobe. The changes were applied within the “Interactive Digital Media” (4002-330) course. Three of the faculty who teach sections of the course met in the summer to explore the new technology. As they became proficient with it they replaced the older (ActionScript 2) modules with newer (ActionScript 3) modules. However the new language was much more capable (stronger object-orientation) and thus it impacted other learning concepts in the course too.

Change should happen at a controlled pace.

> What I did was I slowly made changes to it. One of the problems is that when something doesn't work, there's usually a tendency to go ahead and try to change a bunch of things. And I've tried to not do that. What I've tried to do is always change one dimension of the course on each iteration. Not to try to throw the kitchen sink at it, but here's one thing we can adjust this time. Let's really try to see if this makes an impact or a difference in how the students are doing.

Changes are evaluated at the end of the term in a faculty discussion. In general changes also may be initiated when students and faculty start losing interest in a course.

Some students were involved in the process of exploring the new technology and that gave them a “voice in the process.” Some parts of the design were changed based on their input.

The advantage of using games as class exercises was that they provided a very good vehicle for teaching the necessary concepts. “I could demonstrate multiple objects, managers, moving things around, collision detection, integration of audio and video into the experience, user interface, all that good stuff. And it gave me a vehicle to do it all."

Instructors are necessarily pioneers in developing new coursework. "When you look at curricular design, yeah this was a risk. In looking at our industry partners, looking at where Macromedia itself was going, where we were seeing different things when we looked at industry blogs and things, there was certainly all the indications things were going to go in that direction, and the sooner we figured that out and were able to deliver it to our students, the better." At the time these changes were made there was a scarcity of on-line examples, so the instructors had to develop their own examples.

The labs and the course notes for the course were all changed. The labs didn't change in the sense that the students were still designing a video game, but the underlying technology changed. The change from procedural to object-oriented languages required different decisions and organization of the lab assignments. "The course outcomes never changed. That was the interesting part about it, was the techniques for getting there changed." "All the classroom interactions were now different, because now you're going from procedural to object-oriented. This was a major rewrite, but the outcomes and the objectives stayed the same" The lab assignments in one sense stayed somewhat unchanged, The description from 2004, 2006 and 2008 are all focused on the open-ended assignment to build a game. Some of the terminology changes to discuss objects in the write-up. "But they're very much more geared toward the actual what-are-you-trying-to-do, and not the technology that underlies the issues."

Changes can require significant effort. For example, "

> One of the first outreach things that I did was I made a horrible version of a space, asteroids game, completely object-oriented using ActionScript 2. We did it as a little seminar with a group of faculty and a group of students to say, "Can we make this game object-oriented in Flash using ActionScript 2?" Then we took that, and we folded it into the course, as we revised it in Spring. I spent probably about 4 weeks just coming up with the one demo. And then there was about 4-5 weeks of working with students and faculty to examine that demo, and work through it together, and then an additional amount of time to figure out what parts of the curriculum we could merge it into and see if it would actually work.

Furthermore he stated, "Investment in time: substantial. It's what I love doing, and from a faculty standpoint, we have a chair who is very supportive of these effort. "

The change process requires repeated efforts. The changes from version 1 to version 2 of ActionScript were extensive (described in part in this summary). The changes required in changing from version 2 to version 3 were, "pretty much the same thing."

The teaching approach is "active learning" or studio learning. In the class the instructor works through the design process with example game development and then invites the students to develop their own game.

The institution does expect and require research as well as teaching, but they accept teaching-oriented research. However it is challenging to balance these requirements.

Another example of a course change that was not successful was changing course instruction from polar/Cartesian coordinate problem-solving to using linear algebra and matrices. The problem was aggravated by simultaneously introducing inheritance hierarchies to handle different behaviors for motion (in game design). The combination was too much for the students. They could not internalize and apply those concepts effectively, so the instructor switched back to the polar/Cartesian approach.

Management of grades is done through MyCourses content management system.

End of Interview (Jack)

Interview: Adam: Summary of Interview for Audit (Member Check)

Introductory Remarks. Studying change in courses.
Description of a change: Creation of the IT650, “Computer IO and storage devices” class. (replaced IT528 but was quite different in content).

Motivations for creating it were (1) it addressed the different student interests (IT instead of EET and MET students), (2) it was within your interest/research scope and (3) it could be created in a reasonable amount of time. Since the title of the class is “Computer IO and Storage devices” about 50% of the class is dedicated to each of “Computer IO” and “Storage devices”.

Steps taken to create the class, “I looked at all the computer IO devices and selected from them the ones that I would focus on and then the same for the second half on the storage devices.” Market adoption of technologies and new technologies were important factors in the selection decision. A technology had to be “big enough” I.E. had to have a significant impact on IT professionals. For example keyboards are not interesting, they are commodity devices but biometric identification is more significant.

Peer-reviewed literature and technology industry magazines are key sources for finding new developments to consider for this (and other) classes. Regular reading of the literature leads to making notes in class curricula, which will be adopted into each class on the next cycle.

> “I update my lecture notes every time I read an article that is relevant. I usually read during lunch and if I see something that is relevant I will immediately go right into my lecture notes for next year and just copy this year's lecture notes into next year, change the year, and change them right there. Or if I’m giving a lecture and some student will point something out, I’ll make the note in the margins of my lecture notes and then each year before I teach the class I’ll go back through them and take those notes and incorporate them into the lecture notes”

The structure of the class is kept the same. Technical content within the structure is changed.

Labs play a very large role in the class. (50-60% of total grade). There are three labs (weeks) whose procedures are clearly defined. The remaining eleven lab periods the students independently explore a technology and write up a lab report of it. The topics for the independent labs are proposed by the students and approved by the instructor.

Class time is used to discuss the readings (which the students are expected to read prior to class).

Class management is done through a website (link removed for privacy protection) and grades are calculated using spreadsheets. The Blackboard management system is not used because the user experience is too frustrating.

On teaching style (class interactions): “It's almost always a result of my asking questions or my giving statements that are leading statements. For instance I will say, ‘Here's a CRT. Does that look good? Is that good enough? What is good enough? How do you decide what is good enough?' " To encourage student learning through discussion part of the grade is allocated for Professionalism, which includes both attendance and class participation.

Preferred teaching methods include, "Collaborative learning, toss out a question and field responses to it, give a 2 or 3 minute lecture on something then ask what does that mean to you, expansion. Discussion is what I refer to that as. Guided discussion." And also encourage student involvement because, "An engaged student is a student who will learn; an unengaged student is almost guaranteed not to learn."

Class lectures are periodically enhanced with demonstrations, such as showing how optical fiber can bend light or using beakers of oil to demonstrate refractive indices and so on. Demonstration equipment is also passed around the class from time to time. Field trips are desirable but the logistics are so difficult that they are almost never used.

Another description of a course change - Updating of IT327 digital communications class to include a new concept: The class is taught with a custom edition of the textbook. The textbook is from 2002, which is very old by IT standards but is the best available. The textbook is supplemented with many additional notes and handouts to address newer technologies. The instructor has reviewed many alternative textbooks (about 100) but none had the correct technical focus for university-level IT students, they either emphasize the circuits, as required for EET students or they are targeted at more generally at networking.

The class emphasizes concepts over technologies with approximately an 80/20 mix. Even the labs focus on a concept, not just a technology. The technologies change constantly and are updated by inputs from reading and students as already mentioned. The concepts also change, but much more slowly.

A concept that has gradually grown in importance is Quality of Service (QOS) and a concept that has faded away is Asynchronous Transfer Mode (ATM). A concept that has also grown in importance and had a larger impact on the class is the concept of error correction. Initially this was just mentioned in class with some of the sub-topics (parity, CRC, FEC). As it grew in prominence in the technical literature so the emphasis in class increased. This concept has been around for decades but it wasn't included in the textbook. Initially the instructor followed the textbook, but over time he saw the increasing need for error correction, and introduced it into the class. At some point a lab exercise was created for the students to explore the topic in more depth. Since there are a fixed number of lab periods (14) available one of the other labs was greatly de-emphasized and combined with another lab to create a space for the new lab.

The goal for the new error correction lab was as follows, "I wanted them to actually see data and a parity generated from that data and likewise, CRC. I wanted them to actually pipe data through a CRC generator and watch the CRC sum change as one bit – one single bit out of hundreds – changed. And watch the CRC sum change one iteration at a time. When I did that the first time it was – all sorts of lights went on in my head because I could see how this feedback reads this, and this will never return to that same sum ever again. It will be different because a single bit modified it. That was just to me a very enlightening experience. And I wanted them to get that…" Informal feedback from some students indicates that they also have "really neat" experiences and "Aha moments" in this lab.

This lab is also very concept focused. The students implement one particular technology but as IT professionals they will see many error different correction applications in practice.

The instructor personally created and tested the lab. This particular lab took about six hours to develop. When it seemed ready he handed it off to his TA to see if there were problems with it, before using it in the class. Some procedural problems were identified and changes were made. Generally the instructor personally designs all curriculum changes. Rarely, when an exceptional TA is available, they can be invited to design changes in the lab and then, if approved by the instructor, implement them for the students.

When actually going through a lab experience the students get the instructions from the class website, where the class schedule and lab instructions are posted4. The TA is in the lab with the students providing support as needed.

Change in the IT327 course is incremental and relatively continuous. Each time the class is taught changes will be made to about one fourth of the lectures to introduce new material.

A general comment about change: "What drives [the process of change in IT] is what I hope drives all of our classes, and that is, 'What does the industry need?' Our IAB gives up the feedback as to what's current, or our own readings. If it's no longer current, we don't do it. If it's becoming current and we don't do it, it's time to make a change, and I hope that's the driver." Changes in the curriculum are made, "Because I'm always excited to show them something new. And if it's new, I want them to know it better and understand it. Yeah, it's professional responsibility but that isn't what drives me."

Students can relate to fundamental concepts if they are connected to things they can experience.

> "If you put Shannon's law on the board, it's just a bunch of letters, symbols, it doesn't mean anything. Until you address, okay, this is what's happening in the real world. And here you get some collisions. Now what does it do to your throughput? OK, show me where that appears in Shannon's law? Why is it happening, what's going on?"

Although changes in the course are frequent, changes in lab equipment are relatively rare, because the equipment is versatile and can be adapted to new purposes. Almost all tests are open

---

4

http://www.et.byu.edu/groups/it327/labs.html#Lab%208:%20Error%20Detection%20and%20Co

ding

book, “Because memorization is not understanding. It's one of the lowest levels of Bloom's taxonomy.”

Conclusion of interview. End of Interview (Adam)