Structure of course design in rapidly evolving computing disciplines

Abstract

It is well recognized that computing disciplines need to update their courses regularly due to the rapid evolution of the discipline. These course changes are often difficult, very time consuming and sometimes poorly rewarded by the academic institution, but they are necessary. Sometimes the student learning experience is negatively impacted by the course changes.

A research study was completed to attempt to define the theoretical structural elements of the course change process. Faculty members were interviewed about course changes. Several course change events were identified and these events were analyzed to identify common themes and educational structures so that the process of course change could be better understood and improved.

An analysis has been carried out on a selection of course change events. The analysis reveals the fundamental structure of course change. An in-depth study of a few course change events reveals the theoretical and practical influences that control this change process and allow for future improvements.

The instructional design layers paradigm illustrated some reasons why course changes are resource consuming. Some expected outcomes were confirmed and some new insights were obtained.

An understanding of the abstract layers of design and the conceptual questions addressed by designers in the course design process can improve both the quality of the course changes and the efficiency of the change process.

1. Introduction

It is well recognized that computing changes constantly and rapidly. The influence of exponential growth of computing power driven by Moore's law is often cited as a major contributing influence in this change. The implications of this law are far broader than Gordon Moore's original statement about the doubling of integrated circuit components on a die[1,2,3] and, in general terms electronic systems, particularly computer-based systems, continually and rapidly grow smaller, cheaper and much more powerful Moore's law effectively dominates computer system development.

A consequence of this unrelenting, exponential change is the expectation and need for university instructors to both continually update their own knowledge of the field and also invest considerable effort and resources in updating their technical curricula and laboratories to accommodate these new developments in their field. Therefore instructional design in this environment needs to respond to this continually changing technical landscape.

On the other hand, faculty tenure and promotion decisions in computing disciplines are frequently based on successful research and peer-reviewed publication. Upgrading curricula is seldom well recognized and rewarded for faculty in higher education academic environments. Thus, updating of curricula places additional stress on busy faculty members with little reward other than meeting the needs of their students. This provides a strong motivation to study the process of course evolution, to understand the phenomena involved and to seek improvements.

Furthermore, changing curricula is generally implemented by faculty members trained in computing but with little or no formal training in instructional design. It appears that the change process is a natural and on-going one focused on technical content, with little or no conscious consideration of the impact of on-going change on the structure of the designed educational experiences. Lidtke[4] 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."

As further evidence of the relative lack of attention that is being paid to theoretical design considerations by curriculum designers in this domain, Lister and Box[5] 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 ACM Information Technology Education SIG (SIGITE) conference proceedings (available through the ACM technical publications collection) and another informal review of papers published in the Information Systems Division (ISD) of the 2009 ASEE Annual Conference proceedings supports this conclusion, with the large majority of the published papers focusing on new technology developments that could be introduced into the classroom, a few papers focus on technology to aid teaching and a very few focusing on educational design issues. In other words there is little or no consideration of the educational curriculum architecture required for, or affected by, these constant curriculum changes.

Rapidly evolving technology is not the only influence affecting curriculum change. Other pressures, such as externally imposed requirements for new course management systems or institutional requests for new teaching approaches, among other influences, will also impact evolving curricula.

The problem thus posed is to explore what is happening as Information Technology (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? The longer term goal is to define theoretical structures and practical procedures to improve the instructional design process for computing educators, to enable them to more effectively manage the process of constant change while both ensuring the quality of education for their students is maintained and also either minimizing their time commitment or somehow folding their curriculum development efforts onto the institutionally recognized reward system – typically published research and grants.

The goals of this research project and report are more modest. They are to reliably evaluate some current practices, using recognized educational theoretical bases so that the current situation can be understood in an instructional design context and key indicators of the design process can be identified and that methods of improvement can be based on a correct understanding of the influences at work in the process. A deeper understanding of what is happening will create a foundation upon which future designs or evaluations may be based. Furthermore this report does not report on all of the results of the study, a few of the results were selected to illustrate the nature of the problems and point towards the solutions. A complete report of the research project is in preparation.

Instructional design architecture provides a paradigm for exploring and explaining these problems. If the change process can be examined and interpreted in terms of effects of change on the structure of the design then we will gain a nuanced and meaningful view of what is happening and be able to interpret it in terms of educational theory. This research project proposes using a case-study approach to examine a small cross-section of curricular changes in the IT domain. The results will be 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.

Two questions were addressed in the research: What is the nature and effect of the changes that occur in instructional design architecture when courses in higher-education information technology evolve, and what actions or processes do instructors implement when changing courses?

2 Description of the Research Project

In order to establish a basis for understanding the process of change a modified case-study approach was used. Seven professors from two different four-year universities were identified. Each of them were instructors in Information Technology (IT) programs with varying levels of responsibility and experience on developing, adapting and modifying curricula for IT courses. Examples of course changes were collected from these professors.

The data was collected by interviews with the professors. The interviews were based on a loose set of interview questions in order to elicit natural responses from the interviewees. A qualitative research approach was used, as discussed below.

Direct quotations or *emic perspectives* of interviewees allows the voice of the interviewee to be expressed and is an integral part of qualitative research[6,7] but privacy of interviewees also should be protected. Therefore the interviewees were assigned arbitrary anonymous names so that they could be quoted anonymously. Names of professors referred to in this report are these “anonymized” aliases.

The interviews were transcribed and analyzed and instances of curriculum change were identified. Common themes in the process of change were also sought and identified. Underlying relationships of the change process to theoretical instructional design structures were identified.

2.1 Qualitative Research Study

Different types of research studies serve different purposes. Gibbons and Bunderson in “Explore, Design Explain”[8] suggest three important classifications of research approaches are explaining phenomena, as is typically done in scientific quantitative research, developing and understanding new designs, frequently done in technological research, and, thirdly, exploring a system or artifact, as is done in naturalistic studies. Exploratory research seeks to produce observations that can reveal relationships, suggest hypotheses and categorize observations. This research is an exploratory study. The intent of this study is to explore instances of curricular evolution in IT.

Within exploratory research both quantitative and qualitative methods can be used. In choosing between a quantitative vs. a qualitative approach, Stake[9] 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 this study is concerned with understanding interrelationships, qualitative research methods were used.

Another consideration of qualitative research is the rigor of the study. In quantitative research environments there are well-known mechanisms for ensuring quality in research. Four recognized mechanisms for quantitative research rigor are internal validity, external validity, reliability, and objectivity. Guba and Lincoln[10] identify these mechanisms and propose four “parallel criteria” for qualitative research. These parallel criteria are Credibility, Transferability, Dependability and Confirmability. Credibility is achieved through a variety of techniques for verifying data and findings with sources and other authorities; Transferability is achieved by sufficiently rich descriptions to enable other researchers to validly interpret the findings into their own contexts; Dependability is ensured by various auditing procedures of the research study and Confirmability is achieved by use of best practice procedures and external review. Aspects of these parallel criteria were applied to this research.

Some examples of these parallel criteria that were used include verifying the understanding of the interviews with the participants, documenting (audit trail) the process and changes to the process, using best practices from recognized authorities for procedures, Using established qualitative and educational theories for analysis and confirming results with experts in these procedures.

3 Analysis

Approximately twenty-two curriculum change events were identified from the seven interview transcripts in the study. The number is approximate because some of the events overlap and at least two of the interviewees were both discussing the same technology change event, but from their own viewpoint. Change events of different magnitudes were found, from minor changes of content within a course to complete course re-designs. Not all of these results will be detailed but a selection of them will be discussed.

3.1 Results

Some expected results emerged from the interviews and from the list of change events, such as the fact that all faculty interviewed change their curricula frequently. Some faculty change courses very frequently; one interviewee indicated that he had redesigned or significantly modified seventeen courses within the past five years. 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 faculty were professional full time designers, but were normal university faculty with teaching and research expectations.

Another common and expected theme is that all those interviewed acknowledged the fact of constant technical change, although their responses revealed different attitudes towards the on-going process of technically-driven change. These attitudes range from considering change as enjoyable and a privilege to considering it an on-going duty and even considering it to be a (unwanted) driving force. Some of these attitudes are indicated by the following quotations from the interviews.

Table 1: Attitudes to on-going change

Alias	Relevant quote
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	"I think that in this field especially, an evolving curriculum is just a necessity"
Tom	"IT changes so fast and updates so much that, … you've got to stay up on top of things"
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."
Susan	"I don't think we manage it. I think it manages us. I think we are continually running ahead of the snowball"

A common theme that was not anticipated, although perhaps it should have been, is that faculty members frequently change their courses to suit their personal preferred teaching style. "Lisa" has even created a verb, 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. She also adopted a teaching method that she describes as "Call and Response" but could more typically be describes into terms of question and discussion. At least four of the seven professors interviewed at both institutions redesigned their courses to follow a studio-teaching model. Although professors also incorporated technical changes at the same time as the new teaching model – and attributed the cause for the change to the technological updating, it is clear 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 (or course structure) for some years, even though the technical content in the course will be changed more often.

Another result that emerged from the interviews is that professors design their courses alone. There were very few exceptions to this in the seven interviews and 22 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 educational 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 interviewees were aware of these departments but none made any significant use of them. In one case a few faculty collaborated by meeting together to discuss a new technology that they wanted to adopt. They used the collaboration as an opportunity to learn together but despite the collaboration on technical matters, each professor designed his or her own course. A few of the faculty made use of students to help prepare materials but these students functioned in the role of apprentices, carrying out specific tasks assigned to them. There was only one exception to this, where one instructor allowed an exceptionally talented undergraduate student to design a series of labs. Even in that case the instructor restructured the labs later to fit his personal teaching style and instructional design expectations.

3.2 Structural Analysis

One approach to an analysis of the structural design of the course changes is to use instructional design layers. Instructional design layers define seven different distinct aspects of instructional design. They are inspired by concepts such as Stewart Brand's layers of buildings. Stewart Brand and others describe how buildings evolve over time and point out that the different layers of the building can be changed separately[11]. Stewart Brand's six layers are alliteratively named Site, Structure, Skin, Services, Space-Plan, and Stuff. Brand describes how the different layers age at different rates and 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. So for example the electrical wiring (Services) of a building could be changed without necessarily affecting the other layers. If the electrical wiring is embedded in brickwork (Skin) then both layers will be impacted by a change. In a like manner Gibbons has defined layers of instructional design[12]. His layers are shown in the table below.

Table 1. Instructional Design Layers (from Gibbons).

Content layer	A design must specify the structures of the abstract subject-matter to be taught, must identify the units into which the subject-matter will be divided, and must describe how elements of subject-matter will be made available to instructional functions performed by other layers.
Strategy layer	A design must specify the physical organization of the learning space, social organizations of participants, their roles and responsibilities, instructional

	goals, allocation of goals to timed event structures, and strategic patterns of interaction between the learner and the instructional experience
Message layer	A design must specify the tactical language of message structures through which the instructional experience can communicate content-derived information to the learner in conversational form.
Control layer	A design must specify the language of control structures through which the learner expresses messages and actions to the source of the learning experience.
Representation layer	A design must specify the representations that make message elements visible, hearable, and otherwise sense-able: the media representation channels to be used, the rule for assigning message elements to media channels, the form and composition of the representation, the synchronization of messages delivered through the multiple channels, and the representations of content.
Media-logic layer	A design must specify the mechanism by which representations are caused to occur in their designed or computed sequence.
Data management layer	A design must specify data to be captured, archived, analyzed, interpreted, and reported.

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 noted that 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 we define what Brand refers to in his model as “shearing layers”, IE separating boundaries for different parts of the course design.

When reviewing the events described in the interviews it is apparent that professors did not either consciously or unconsciously attempt to separate layers of the design to allow for future changes. There were two notable exceptions to this generalization. Firstly in several cases the course outcomes were deliberately not changed while the technology of the course was changed. This was partially an attempt to minimize change 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 approach (Strategy Layer) and then (nominally) the strategy was held constant for future changes in the Content Layer (new technologies etc.). However these attempts, while well-intentioned, did not achieve their desired effect.

Firstly the attempt to keep the course outcomes stable shows an inherent awareness of the problem of multiple layers as aspects of design but in some cases the problem persists and is merely hidden. This was highlighted by the following example from an interview with “Geoffrey.” They have updated a two-course programming sequence as the programming languages used evolved, most recently moving to an outcomes-oriented language. In his own words he describes the results.

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 (from "Geoffrey" interview)

This situation, of the course changing substantially, occurred in a course whose list of outcomes remained unchanged while they updated the technology they were using. The course changed from one type of programming experience to a quite different type of experience moving to an object-oriented language, but the course outcomes were defined in terms of simple programming constructs (variables, loops etc.) and so did not change. In this case there is a significant disparity between the nominal (listed) outcomes and the effective outcomes of the course that is not being acknowledged, which leads to this "worst-case scenario."

In the second circumstance the professor changes the course strategies to suit their teaching model 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 changing a course to what one instructor described as a "Call and Response model" (question, challenge and discussion, with exercises – a variation on a studio teaching model), meant that not only did the strategy layer change but the control and message layers changed too. IE 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 the course is taught by a different instructor 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. The professor acknowledged this problem by indicating that a when different colleagues teach 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 interviewee "Lisa" Edited from verbal transcript to anonymize names and for clarity.)

Continuing the analogy of Brand's building layers this is similar 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 a different space constraints. While this leads to very individualistic designs it is very inefficient and leads to haphazard changes to the system.

The design layers analysis approach proves to be a powerful tool in illustrating the structure of the problems that can arise in instructional design in the process of change.

3.3 Example of change

Although a number of changes in courses were documented it is revealing to discuss a single 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. 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, which aligned better with other object-oriented languages such as Java, C# and C++, which 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 beta-testing of systems before because of this close relationship with Adobe updating the language is an important goal for them. These benefits all relate to the benefit of students using specific current technologies. The adoption of this language as an introductory programming language did lead to some smoothing of the 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 four years and have done similar major changes in the past, it can be reasonably surmised that they will go through similar major changes again in the future as the technology changes again.

An interesting fact in these major changes is that the (nominal) course outcomes did not change substantially. The professor stated that course outcomes had been defined in terms of fundamental programming concepts and graphic animation tasks and not in terms of a particular language. The conclusions I draw from this is that defining your outcomes in a technology-independent way does not necessarily protect you from major efforts in re-design to keep up with new technologies. Also, as described above, carefully designed technology-independent course outcomes may vary from the actual outcomes in ways that can have a significant impact on students and departments.

Another interesting factor is that the organization 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 instructors 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 75 hour week or more. This particular professor seemed to accept that changes are both desirable and necessary but felt that the need to change was driving the professors rather than the other way round. IE the professor felt considerable pressure to keep up with the changes in the light of all her other responsibilities.

This periodic large-effort change in curriculum I have characterized as the "heroic conquest" model. Professors spend great effort completely revising the curriculum on an occasional basis – say during the summer every couple of years. In contrast to the heroic conquest model a few professors use what I refer to as a "steel girder" model. They redesign the course to their preferred teaching style and thereafter change domain-specific technical content fairly continuously in small increments, but without disturbing the structure of their instructional design. The disadvantages of the heroic conquest model are obvious. The advantage is that course content and structure can be rebuilt for new technologies of new teaching methods as necessary. The steel girder model avoids the huge periodic effort of rebuilding but lacks flexibility and adaptability.

This Actionscript updating experience can be contrasted to another one identified in this study. One of the professors described how she modified the classes she taught to suit her preferred style of teaching. She indicated that she worked in a graphic-art oriented sub-section of the program. 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 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 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.

Another factor of interest in this study was to what extent professors made use of external resources 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. A few of them used student help to create materials but in all but one of these cases the students were just implementing tasks clearly defined by the professor. In the exceptional case the professor was working with an exceptionally talented undergraduate student and allowed that student to define a series of labs and create working prototypes of them. Even in that case the professor reworked the labs to suit his own model of instruction.

4 Lessons Learned and Recommendations

Some of the expectations of this study were confirmed. Professors are continually updating their courses, they recognize that the technology of the field is constantly changing and, on the whole, accept the need to continually incorporate changes into their curriculum. Also in these interviews the professors indicated that change can consume significant time and that the institution does not strongly value these efforts in terms of promotion or financial rewards. They are willing to stay current in their discipline but find that continually modifying their curricula is necessary, resource-consuming, expected but unrewarded.

When it comes to methods for incorporating changes into the curriculum, more variation was seen. Some faculty accept the considerable extra workload as a necessary part of their employment, others have managed to streamline the process by slowly and incrementally adjusting their curriculum in an orderly manner. These professors have effectively frozen multiple layers of the instructional design and they just make adjustments to one of the layers (usually within the content layer)(steel-girder model). This strategy works but at the price of possibly stifling innovation in instructional design. Other professors invest significant amount of time and other resources in intense bursts (EG in the summer), and redesign many layers of the course at the same time (heroic conquest model). Some regard it as more of an ad-hoc process and use unstructured approaches to change. A recommendation for future work in this area is that if professors understood the layered nature of instructional design they could more thoughtfully identify shearing layers in their design to give themselves the freedom to adapt as necessary but not to interfere with multiple layers of the design. On-going research is looking at methodologies to make the layers visible and useful during the curriculum design process and to inform professors of the need and benefits of this approach.

One glaring fact noted in this study is the predilection of professors to work as individual craftsmen, or to cooperate with technical colleagues teaching related courses but not collaborate with colleagues with a background in course design or with on-campus faculty support departments especially set up for this purpose. Each course is an individualistic, even artistic, creation by a single expert, the professor. Nothing is going to relieve the faculty of the necessity of remaining current in their field and several professors indicated their desire, even their delight, in doing so, however there is no reason why faculty cannot either seek out instructional designer expertise in finding new ways to approach instructional. It appears that course designers do not recognize the problems inherent in solo design by individual experts. Universities are providing resources and it is probable that education departments on campus would also be willing to collaborate. Course designers need to recognize the benefits of a shared effort, preferably based around best instructional design practices to modernize the process of instructional design.

The clearest recommendation that can be made for future improvement at this stage is to be aware of the layered nature of instructional design. Attempts to keep course outcomes static are a move in the right direction but designers need to be aware that course outcomes must describe the real intents of the course not just a convenient sub-section of them. Future research in this area will explore effective ways of implementing this approach.

Bibliography

1. Brock, D.C. and G.E. Moore, *Understanding Moore's law : four decades of innovation*. 2006, Philadelphia, Pa.: Chemical Heritage Foundation. 122 p.
2. Moore, G.E., *Cramming more components onto integrated circuits.* Electronics, 1965. **38**(8).
3. Schaller, R.R., *Moore's Law: Past, Present and Future.* IEEE Spectrum, 1997. **34**(6): p. 52-59.
4. Lidtke, D.K. *What's new in curriculum design: working with industry*. in *28th Annual Frontiers in Education Conference*. 1998.
5. Lister, R. and I. Box, *A citation analysis of the SIGCSE 2007 proceedings*, in *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. 2008, ACM: Portland, OR, USA.
6. Vidich, A.J. and S.M. Lyman, *Qualitative Methods: Their History in Sociology and Anthropology*, in *Handbook of Qualitative Research*, N.K. Denzin and Y.S. Lincoln, Editors. 2000, Sage Publications Inc.: Thousand Oaks, London, New Delhi. p. 37-84.
7. Williams, D.D., *Educators as Inquirers: Using Naturalistic Inquiry*. 2009: Provo.
8. Gibbons, A.S. and C.V. Bunderson, *Explore, Explain, Design*, in *Encyclopedia of Social Measurement*, K. Kempf-Leonard, Editor. 2005, Elsevier: New York. p. 927-938.
9. Stake, R.E., *The Art Of Case Study Research*. 1995: Sage Publications Inc.
10. Guba, E.G. and Y.S. Lincoln, *Fourth Generation Evaluation.* 1989, Newbury Park, London, New Delhi: Sage.
11. Brand, S., *How Buildings Learn: What Happens After They're Built*. 1994, London: Viking Penguin.
12. Gibbons, A.S. and P.C. Rogers, *The Architecture of Instructional Theory*, in *Instructional Design Theories and Models: Building a Common Knowledge Base (Vol III)*. 2009, Routledge. p. 305-326.