Promoting Creativity in the Computer Science Design Studio

Katherine Cennamo
Learning Sciences and Technologies
Virginia Tech
Blacksburg, VA
cennamo@vt.edu

Sarah A. Douglas
Computer Science
University of Oregon
Eugene, OR
douglas@cs.uoregon.edu

Mitzi Vernon
Industrial Design
Virginia Tech
Blacksburg, VA
vernon@vt.edu

Carol Brandt & Brigitte Scott
Teaching and Learning
Virginia Tech
Blacksburg, VA
cbrandt@vt.edu & brigit2@vt.edu

Yolanda Reimer
Computer Science
University of Montana
Missoula, MT
reimer@cs.umt.edu

Margrita McGrath
Architecture
Virginia Tech
Blacksburg, VA
mmcgrath@vt.edu

ABSTRACT
Revolutionary advances in technologies will require computer science professionals who are able to develop innovative software solutions. In order to identify techniques that can lead students to creative insights in their work, we have conducted an ethnographic study of the studio method as enacted in architecture, industrial design (ID), and human-computer interaction (HCI) classes. Our analysis of the activities conducted during studio critiques revealed that while the ID and architecture studios had a primary focus on experimentation, the primary emphasis of the HCI studios was on idea refinement. In this paper, we describe four barriers to creative thought observed in the HCI classrooms and identify ways that the architecture and ID instructors helped students to overcome similar challenges.

Categories and Subject Descriptors
D.2.2 [Software Engineering]: Design Tools and Techniques H.5.3 [User Interface]: User-centered Design

General Terms
Design

Keywords
Design, studio, creativity, computer science education research

1. INTRODUCTION
“Designers put things together and bring new things into being, dealing in the process with many variables and constraints, some initially known and some discovered through designing.” [15 p. 42]

Design is prevalent in many areas within computer science. For example, application areas such as graphics and visual programming, artificial intelligence, information systems, and human computer interaction all may require the design of algorithms, interfaces, interactions, programs, specifications, simulations, and/or systems. Many years ago Herbert Simon [17] described design as an instance of problem solving, moving from the current problem situation progressively closer to the desired goal situation. Problem solving was simply a matter of finding a workable solution to a designated problem. In contrast, Frederick Brooks [1] argues that the Simon rational model is not sufficient to describe real-world design activity. Based on his experience as a software engineer, Brooks believes the hardest part of design is deciding what to design. The development of innovative hardware and software solutions often requires designers to solve problems that are “at least in part, uncertain, ill-defined, complex, and incoherent” [18, p. 42]. Although we need software designers who can follow rules when presented with technical and rational problems, we also need designers who can make good sense out of those problems that are not technical or rational: that is, designers who are aware of multiple possibilities for solutions, who can make good choices, and who can reflect on the choices they make to determine if their goals have been met.

During the past 20 years, the goals of a human-computer interaction (HCI) curriculum have changed from primarily teaching the psychological foundations and surveys of human-computer interaction technologies to teaching competence in designing interactive objects. As early as 1990, Professor Terry Winograd of Stanford University in a CHI conference plenary entitled “What can we teach about Human-Computer Interaction” urged the participants to embrace HCI design education and explore the studio as a teaching concept [22]. Since then, several innovative computer science programs have incorporated the studio approach within their curricula (e.g. [4, 7, 8, 14])—albeit the logistics and procedures of the design studio have been described for HCI education [8, 14]—project-based assignments, supplemented with public critiques and individual student-teacher conferences—little is known about the instructional methods and processes through which students can learn to develop innovative design solutions.

As indicated in the National Science Foundation’s Science of Design call for proposals “Revolutionary advances in hardware, networking, and human interface technologies will require entirely new ways of thinking about how software systems are
conceptualized, built, understood, and evaluated. ... To these ends, importing and adapting ideas from other design fields (engineering, biology, architecture, economics, and the arts, for example) are encouraged” [12, p.2].

The central goal of this project was to examine studio-based courses as they are enacted in the traditional design disciplines of architecture and industrial design (ID), as well as in HCI, to discover ways in which students learn to develop innovative design solutions. To achieve this goal, we are engaged in a 3-year research and education project involving 6 faculty across 3 universities representing HCI, ID, architecture, learning sciences, and qualitative research methods. In this paper, we focus our analysis on how studio-based activities guide students to creative insights in their work.

1.1 What is Studio?
In the traditional design studio, students work in an environment where they are provided with dedicated workspace available to them at all times. Studio classes usually meet three times a week for several-hour sessions, but students are encouraged to work in the studio rather than at home during off-hours.

The specific teaching methods vary by program and university, but typically, students are presented with a design problem that is grounded in the realities of professional practice. Although periodic lectures on some aspect of the design problem may be delivered as needed, students are expected to work independently or in groups to learn various skills from each other. Faculty and peer interactions stimulate student reflection on, and discovery of, their developing knowledge. At various points in the class, students present their work to faculty, peers, and occasionally guests for critiques that complement and extend informal reviews. The project critique, or “crit”, can take many forms including desk crits, pin-ups, juries, reviews, and open houses [3].

As enacted in computer science education, the design crit is the cornerstone of the studio method [8]. In her examination of the role of dialogue in a design studio, Dannels [3] supports the prominence of the design crit and argues that the oral genre of the crit is the site for learning disciplinary conventions and for developing a sense of oneself as a professional.

1.2 What is Creativity?
In some fields, “creativity” is valued through aesthetic expression. However, creative thought is also of value in domains focused on problem solving, such as HCI. Whether the goal is to develop an aesthetically pleasing product or to solve a challenging problem that requires thinking in new ways, the literature on creativity and innovation reveals that development of creative solutions is an iterative process, involving a cycle of idea generation and refinement.

The ability to generate a wide variety of potential solution ideas is generally accepted as a key precursor to creative design outcomes. In fact, most measures of creative thought assess creativity in terms of how many possible solutions are generated to a given prompt, as well as the originality of the solution in terms of how different the solutions are from one another (e.g. [9, 20]).

Researchers in the area of creativity (e.g. [2, 12, 18]) have identified several techniques that contribute to the generation of multiple original ideas. Creative thinkers are able to examine ideas from various perspectives, often exploring a challenge using a variety of raw materials, stimuli, and experiences to provide alternative perspectives on a challenge. As they explore ideas from various perspectives, they observe carefully, make inferences, and elaborate on their thinking. In addition, creative thinkers use analogies or metaphors to think through novel problems, reasoning from examples or similar situations to consider multiple possible courses of action.

Although many measures of creativity focus on the fluency and originality of ideas generated in response to a problem or prompt, the development of creative products also requires reflective judgment to select the best ideas to move forward. Reflective judgment requires analysis, synthesis, and evaluation skills [2]. The studio crit is, to a great extent, intended to develop such skills. Students present their work to faculty and other students. In order to do so, they organize and summarize their work in a way that communicates their ideas to others. They interpret their work and respond to questioning. And during this process, they learn to evaluate their work and the work of others through making judgments about the value, worth and logic of the product as well as its applicability to the problem situation.

However, it must be recognized that this is an iterative process. Creative thinkers do not simply generate a variety of ideas, and then select the best one to move forward. Instead they cycle through the process multiple times—generating ideas, judging those ideas, using the outcomes of their reflective judgment to generate more and different ideas, and so forth until a desired solution is reached.

2.0 METHODOLOGY
The purpose of this analysis was to examine the studio method, as enacted in architecture, ID and HCI classes, to identify how the creative design process unfolds and techniques used to encourage creative thought. This research uses ethnographic case study methodology as described by Merriam [11, p. xiii], in which a case is defined as “an intensive, holistic description and analysis of a bounded phenomenon such as a program, an institution, a person, a process, or a social unit.”

The present study did not involve a purposefully designed intervention, but rather focused on the everyday practices of the studio-classroom community. Data have been collected and analyzed from one semester long course in architecture, one semester long course in ID, two semester-long courses in HCI, and one quarter-long class in HCI.

2.1 Context of Investigation
The architecture and ID studio classes met for 4-hour blocks of time, 3 days per week. In addition, each student had a dedicated workspace in the studio accessible 24 hours a day, 7 days a week, 365 days a year, and students were encouraged to work in the studio space outside of scheduled class time. Students were presented with a series of assignments that they were expected to submit individually; however, they were encouraged to use additional resources, including their peers, as needed to complete the assignment. Project-based assignments were supplemented by frequent public critiques where students presented their design ideas to faculty, peers, and occasional guests. Students in both courses were in the first semester of their second year of professional study and had participated in a two-semester studio experience the prior year.
The three HCI courses that are the subject of our investigation were selected because of the prior experience of the two course instructors in implementing the studio method into HCI instruction. These instructors have used a “modified studio approach” based on their observations of an architecture studio [14] for at least ten years. A modified studio approach was necessary because the typical course structure of computer science departments made it difficult for all of the features present in architecture or ID studios to be implemented easily. As is typical in studio classes, the HCI courses incorporated a series of project-based assignments followed by design crits where students publicly presented their concepts to their peers and professors. Approximately half of the class time was devoted to lectures, with the course assignments and associated crits providing an opportunity for the students to apply the principles that were introduced in the lectures. Class projects were completed in teams. These courses did not provide dedicated studio space for students, nor was the studio scheduled for the extended hours that are common in architecture and ID studio courses. Instead, students met in a typical classroom for approximately three hours per week, as they would for a standard lecture-based course. The courses were each elective, enrolling students who were at the senior or graduate level. Thus, a key aspect of our work was to identify ways to facilitate creative design thinking within the constraints of the computer science academic environments.

2.2 Role of Researchers
The authors of this paper represent the two educational researchers, the research assistant, and the instructors of each of the courses. Data were analyzed by the educational researchers and assistant while the course instructors, as key informants, reviewed the cases to provide domain-specific insight on the educational researchers’ interpretations of the data.

2.3 Data Collection and Analysis
In all the classes, key classroom interactions, as identified by the instructor of each course, were videotaped for analysis. In addition, we collected all student and instructor-generated artifacts produced for each class as well as instructor reflections on the course activities. The bulk of our data was captured through video which included: project crits, discussions among students in project groups, and discussions between faculty and students (at “roundtables” or “desk crits”).

Each of the five courses served as one case study [11]. Each case was analyzed within case for elements essential to the studio approach. Constructing the cases for each studio course relied upon qualitative data analysis in which the researcher undertakes a continual process of looking for patterns in themes and categories within the data. LeCompte [9] describes this method of qualitative analysis as a series of step-wise activities. This technique also is partially derived from the tradition of constant comparative analysis as described by Glaser and Strauss [6].

To understand the cultural practices and construction of knowledge in the design studio, we developed tools to track and analyze the process of learning over the course of a university term (semester/quarter). Each video was viewed in full at least two times, with sections of the video watched multiple times for a more detailed analysis of dialogues and action. While watching the video, the analyst composed a written narrative marked frequently with time stamps. This written narrative provides a detailed description of the action in the video by topics, speakers, objects, and contexts.

The construction of the cases began by first organizing the video narratives and other diverse types of data for each course into a case. In order to organize our data, we relied on the coding categories of surface features, pedagogy, and epistemology, derived from Shaffer’s [16] analysis an architecture studio at MIT. After several readings of the data, and several drafts of each case, we developed a tentative version of each case for instructor review.

Lincoln and Guba [10] coined the term “trustworthiness” to talk about reliability and validity in qualitative research. Our attempts at trustworthiness focused on gathering a wide range of data that has assisted us in being able to see alternative explanations and other dimensions to the analysis and construction of the research text. We collected a range of different types of data over an extended period of time (an entire semester or quarter) that included videotapes of interactions in the studio, artifacts (student assignments and final projects), syllabi, project briefs, instructor journals, and student surveys. These various kinds of data “triangulated” or corroborated our initial conjectures. We brought our final narratives of each case back to the instructors for member checking. In some situations, we went back to the video or primary data to double-check those places in the case narrative highlighted by the instructors.

Once the cases were finalized, the individual cases were reviewed and coded for factors that have been shown in the literature to facilitate creative thought and innovative outcomes [3]. Excerpts from the cases were coded as to whether they focused on idea generation or idea refinement, and within those broad categories, the researcher further coded the instances using categories derived from the literature. In addition, the five cases were analyzed to discern patterns within and across studio classrooms. The findings we report were informed by multiple data sources and were observed, in one form or another, on multiple occasions and in multiple classrooms.

3.0 FINDINGS
One key difference between the Architecture and ID studios we observed and the HCI studios observed was the focus on idea generation relative to idea refinement. While the ID and architecture studios have a primary focus on experimentation and innovative thought, the primary focus of the HCI studio activities is on idea refinement.

Our discussion of the findings will be organized by four primary challenges to creative thought that we observed in the HCI classrooms and techniques used by the ID course instructor to push students to generate new and different ways of approaching the design problem. Although we use excerpts from one ID course and one HCI course to illustrate our findings in order to streamline the discussion, the trends observed in the HCI course we describe were observed in the other two HCI courses and the trends observed in the ID course were evident in some form or another in the architecture course as well.

3.1 Understanding the Place of the User’s Experience
In all the studios we observed, the faculty encouraged students to see a design problem from the perspective of the user. For
example, HCI students were asked to perform a walkthrough of a user interacting with a peer’s design and ID students were asked to imagine the user’s experience. However, in most cases, these activities were in the service of different goals. For example, the second exercise in an HCI class required students to put themselves in the situation of actually trying to use the software the students designed while blindfolded. Students were either “blindfolded testers” or observers. The activity in which students attempted to work with an interface while blindfolded was intended to help students see the problem from a different perspective, but still, within the context of evaluating a proposed solution.

In contrast, the ID instructor used a focus on the user’s experience to push students to move beyond obvious solutions in the initial stages of idea generation. For example, when refining designs for a medical hand dispenser and “brainstorming” with the students, the instructor created a narrative in which she was a doctor or nurse going about her day and thinking out loud the various ways in which she might use this dispenser, how the dispenser could be used for various tools, and how easy or difficult it would be. By thinking and surmising out loud, she drew the students into the project on a deeper level, expanding their perspectives, and engaging them in the problem of the design.

### 3.2 Breaking Beyond the Familiar

Students often relied on familiar templates for their designs. One group of HCI students, for example, borrowed models from iTunes to design their user interface. In her critique of their final project, the instructor encouraged students to expand their design options: “One of the problems of using an analogy like iTunes is that it’s so well-developed that you can’t get your head outside of it. And you want to try and burst through that”. But despite the instructor’s exhortation, students seemed unsure how to begin to design a novel HCI interface.

Students in ID also relied on familiar ideas in their initial designs. As students were refining their designs for the medical hand dispenser, one student was trying to develop a non-threatening alternative to a syringe for children’s vaccinations. The student developed a series of toy-like sheaths that could be fitted over a syringe, which generated much discussion among his classmates. At this point, the instructor stepped in to actively model the syringe, which generated much discussion among his classmates. The activity in which students attempted to work with an interface while blindfolded was intended to help students see the problem from a different perspective, but still, within the context of evaluating a proposed solution.

Students in ID also relied on familiar ideas in their initial designs. As students were refining their designs for the medical hand dispenser, one student was trying to develop a non-threatening alternative to a syringe for children’s vaccinations. The instructor created a narrative in which she was a doctor or nurse going about her day and thinking out loud the various ways in which she might use this dispenser, how the dispenser could be used for various tools, and how easy or difficult it would be. By thinking and surmising out loud, she drew the students into the project on a deeper level, expanding their perspectives, and engaging them in the problem of the design.

### 3.3 Overcoming Limits

When generating ideas, the students in HCI often limited their solutions to what they could program. An example of how skills and experience with programming served to limit students’ designs occurred early in the semester during the second project crit. Students were presenting a sprinkler system, and a volunteer user was attempting to set up the system. The volunteer became confused because the program did not contain a pop-up window that the group knew needed to be there but that they could not program:

Student 1: I have taken a mental note. That was actually a part of our heuristic evaluation, was that we needed a better…

Student 2: (interrupting) It was in the primary design that we would use the pop-up window, but I couldn’t make it in html.

Instructor: But again, I want to stress, right, in these early designs you should not be limited in terms of what you show us based on what you can program. . . . We’re not even talking about implementation. These could just as well be a hand-drawn sketch. So if there’s something that happens, you could just put up a little hand-drawn pop-up to give us some feedback. But I don’t want your designs to be limited at this point by what you can or cannot program.

Despite the instructor’s encouragement to avoid such limitations, students’ awareness that they would eventually have to build their design created a tension between innovation and what they could build.

In order to overcome students’ tendencies to focus on a narrow set of achievable solutions, the ID instructor often required students to generate multiple ideas that would then be winnowed down through a public discussion. The ID instructor noted in her journal that: “So, I decided to use a method of iteration that I had used before with juniors as a way of giving structure and a deadline, which is often so much more valuable than time. We asked the students to develop 10 different concepts for their dispensers on 10 different pieces of paper…We spent much of last week going through these pin-ups on the wall.” Pinning up all the designs next to each other permitted students to view the design problem from alternative perspectives. This process helped students to make decisions, but also demanded that they begin the design process by generating a broad range of ideas, some of which pushed the boundaries of being practical. This emphasis on innovation
permitted students to have the freedom to think broadly and to experiment with their ideas.

3.4 Need for Low-fidelity Work

We also observed that the HCI students’ willingness to modify their designs was influenced by the materials used to develop and present their ideas. As designers of computer programs, the use of computer technology, often PowerPoint, was a dominant way of communicating their ideas in the project crits. Students created annotated screen shots and detailed digital mock-ups of interfaces, as well as presenting the typical bullet points and scanned drawings. However, the media through which students executed their initial ideas influenced their willingness to modify their designs. The instructor encouraged students to avoid a polished creation in the early stages of the design process saying: “Detail here is not critical. This is the whole reason I urge people to take poster paper and just rip it up. As soon as we make it really beautiful and put it on PowerPoint slides, you become very invested in this design and it becomes very difficult to throw it away”. Yet students appeared to be at a loss as to how to communicate their designs through other media.

In the third exercise, the HCI instructor asked one of the groups to draw their design on the white board, rather than relying on PowerPoint, or in this particular case, very small index cards. As the student drew the display and then the controls, it created an interactive situation: students were asking the presenter questions as he was creating a context for his project, as well as clarifying what the actual drawings were. At this point in the crit, the instructor stepped back and students guided the discussion.

After this crit, the groups began to use the white board and function as a team where one person would narrate key design features while the other team members would sketch design features on the white board. When students presented their original ideas in a polished form, they were less open to change. When the same students made hand-drawn representations, the discussion was richer and fostered more interaction among classmates.

In the ID studios, multiple modes of conceptualization were encouraged constantly. We observed that not all students work well in every form. Exploring design ideas using a mix of tools such as computer-aided devices (software, scanners, 3D printers, digital photography) and analog methods (3D physical hand-built modeling, hand sketching, workshop construction) elicited fluency, originality, and iteration, hence affording a greater probability of reaching the best resolution.

4.0 DISCUSSION

As an ethnographic study, we do not propose that our findings are necessarily true for other HCI, ID, or architecture studio classes. Yet, we believe we have gained a deeper understanding of ways to develop creative thought through our analyses.

It could be argued that our conclusions may be an artifact of the data we collected. By collecting data of the project crits, were we indeed focusing on idea refinement rather than generation? Yet the project crits provided ample evidence of the encouragement of divergent thinking within the ID and architecture studio courses.

There may be many reasons for this obvious difference, not the least of which is the differences in the time available for studio work. Whereas the ID studio had 172 hours of time dedicated to formal studio activities, the HCI classes formally met for only 37 hours in the semester-long classes and for only 24 hours in the quarter-long class. The severe constraints of time within the HCI studio were evident throughout our observations; students and faculty constantly struggled with the time restrictions.

Yet even in the most time-limited HCI course we observed, the instructor was able to make changes in her course activities that positively affected students’ openness to new and different ways of seeing, once she became aware of the need to focus on experimentation and ideation, as well as evaluation and refinement. Based on her review and reflection on our studio case studies, she made several simple changes that had a positive impact. An analysis of videotaped data collected during the formative evaluation of her modified course revealed the impact of one of these changes.

The seemingly simple requirement that students may not use PowerPoint as a mode of presentation during project crits made notable changes to the HCI classroom environment. In the prior data collected in this HCI classroom, PowerPoint served to be a severe limitation to students’ willingness to modify their design ideas. As the instructor mentioned to her class, taking the time to put together a Power Point presentation only serves to make students feel highly attached to their designs—and less likely to be open or amenable to critiques and suggestions that would necessitate design changes. The instructor insisted from the outset that students present their projects using hand-drawn illustrations, whether on paper or on the white board. The impact of this simple change was powerful in comparison to previous HCI courses for which we have data.

In lieu of performing project crits using PowerPoint, students instead either drew their interfaces on the white board in the classroom while they talked (often manipulating, erasing, redrawing, drawing arrows, and so forth), or taped hand-drawn, usually poster-sized paper illustrations to the white board while presenting. As students physically engaged with a static, imperfect drawing, they seemed to naturally present their work as “in process,” talking and making additional drawings as they went along, in order to narrate their thinking. As Tufte [21] has noted, PowerPoint tended to encourage students to linearly narrate their designs—from start to finish. The difference between narrating “thinking” and narrating “designs” was also a function of placing emphasis, respectively, on process and product. The conversations these drawings produced seemed, in comparison to previous data collected, to be quite additive, meaningful, and productive. The students presenting their designs were very receptive to their peers’ comments, and likewise, the students seemed very comfortable and free to provide a critique knowing that the presenters, by way of their presentations, were soliciting comments to help move their designs forward.

Whether designing algorithms, interfaces, interactions, programs, specifications, simulations, and/or systems, students can be
encouraged to generate multiple design ideas before winnowing their ideas down to the one that will be implemented. At least in the initial stages of idea generation, computer science instructors may want to separate design from implementation, possibly by loosening the requirement that all—or even most—designs eventually get implemented. In the HCI classes we observed, one very effective technique for evaluating designs involved students putting themselves into the role of the user in order to switch perspectives and experience how their own designs might be interpreted and experienced from different viewpoints. Students can use this technique during the initial generation of design ideas as well; carefully considering the design problem from multiple perspectives can effectively open up new possibilities for productive solutions. Furthermore, the role of the instructor in modeling how they would go about generating multiple innovative design solutions is of critical importance in teaching students how to go beyond the limits of their own programming skills and in thinking beyond familiar designs.

5.0 CONCLUSIONS

The studio method is gradually becoming an accepted method of teaching design skills to computer science students. However, if we want future computer science professionals that are able to think beyond the obvious solutions to develop innovative products, we need to use assignments and design critiques for more than idea refinement. Students learn to value the things we emphasize in our courses through activities and project assessments. If we want our computer science students to become innovative problem solvers who go beyond the obvious to develop creative solutions to novel problems, we need to find ways to value novel solutions in our course activities. From our analyses, we see that projects and associated critiques can be used to encourage students to think broadly and engage in divergent as well as convergent thought, so that, as designers, they are able to creatively “put things together and bring new things into being” [15, p. 42].

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation (NSF) Grant No. 0725290. Opinions expressed are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES