EVOLUTION OF DIGITAL DESIGN TEACHING: A COURSE AS MICROCOSM FOR EDUCATIONAL ISSUES

Nancy Yen-wen Cheng, nywc@darkwing.uoregon.edu

Editor's note: This paper was previously peer-reviewed by ACADIA and presented at the ACSA National Conference 2001 in Baltimore, MD.

Abstract

Despite widespread use of computers in the architectural profession, computer use in architectural education remains uneven. The challenge to educators becomes apparent in examining the evolution of an introductory course. In four years, the teaching initiatives illuminate the crucial issues:

- Content focus (what): computer techniques supporting design concepts, selection of design and communication applications
- Delivery techniques (how):
  - Organizing framework: staffing, course format
  - Teaching tools: web resources, online bulletin boards, online quizzes and gradebook.

These efforts have produced gradual progress. Major successes include development of successful assignments and resources, balance of exercise types, and skill improvement through competency exams. On the other hand, addressing different skill levels, providing personal attention in an efficient way and overcoming equipment impediments remain a challenge.

Outside the course, the overall curricular framework needs to be adjusted to prepare for and reinforce learning within the course. Results from initiatives inside and outside of the classroom are discussed.

Introduction

Architecture schools are now more than ever being challenged to adapt to changing circumstances. While schools have always had to prepare students for today's profession, now we need to anticipate needs for tomorrow's profession. Construction materials and methods are evolving, practice structures are becoming more adaptable, and design trends flow around the world at Internet speeds. In a world filled with change, students even expect history to be taught with high-tech methods. But academia operates at a slower institutional pace, budgets may curtail upgrading equipment, and faculty may be isolated from the parts of society they serve. While all professional education must deal with increasing specialization and growing bodies of knowledge, the computing aspect of each curriculum is most characterized by constant change. Technology's accelerating curve accentuates the challenge of integrating relevant computing skills into architectural education.

By looking at the evolution of one required introductory course, it is possible to see the challenge close-up. Two major issues have shaped the evolution of the course: defining content and engaging students. In choosing what to focus on, there is a trade-off between depth and breadth. In engaging students in this content, both the organizational framework and teaching methods have been manipulated. Teaching initiatives in each of these directions have yielded incremental progress. Putting these initiatives in the perspective of larger curriculum planning makes it possible to understand the major issues and strategies for dealing with them.

Background

Background context of the course explains part of the teaching challenge. The faculty at the University of Oregon recognized early that computing would be playing a growing role in the profession, piloting computer studios in the late 80's. In 1995, it was one of the first architecture programs to require students to purchase their own computers. Concurrently, the faculty decided all students would be required to show computer literacy in image processing, 2D drafting, 3D modeling and rendering or take an introductory course to cover these topics. This required introductory course was set in the fall quarter for all entering students so they would learn computing prior to being acculturated into manual media. It was envisioned that computing would permeate all aspects of the curriculum, just as design thinking does. This intention shaped faculty hiring practices: Of the nine tenure-track faculty hired since 1994, all have some degree of computer literacy and five require computer aided design, computer graphics or web authoring in their classes. The department added a second full time position in 1996 and an additional part time position in 1998.

Since that time, our department has expanded the offerings in computer media to include several sections of construction drawing, two advanced media courses and two intermediate studios per year. Advanced topics have included lighting, animation, web programming, advanced modeling & rendering or design collaboration. The department has made the transition from only Macintosh to one that supports Windows and Macintosh equally well. There has been notable growth in computer analysis in energy and environmental courses and the use of desktop publishing, web-authoring, and computer graphics in both studio and subject area courses.

Still, the only required computing courses are merely 4 credits of a 241 credit five-year Bachelor's degree and 2 credits of a 144 credit three-year first professional Master's program. While a small group of the students develop their computer skills throughout their studio sequence, the majority of students take only the required course and with skills reinforcement in the core studios.

Content Choices: what the course is about & how it has changed

Given that many students' only exposure to computing technology comes in this course, the choice of what they learn becomes very crucial. Pressures to learn currently desirable practical skills can crowd out more timeless fundamental concepts. In order to understand the skills in a more meaningful way, students must also be exposed to their context including: the range of computational techniques and directions for the future. From the many possibilities, I have chosen to concentrate on design concepts and how computer operations and structures support them. Students learn about principles and processes in lecture followed by hands-on skill development in computer lab tutorials. While the design concepts are featured as the motivation (as in Madrazo 1998), the course is structured around exposure to constructs in the software. In the most recent iterations, the course has focused on web communication, geometric transformations, kit of parts construction and top-down abstraction and refinement.

In choosing what to teach during a 11-week quarter, my teaching partner Scott Howe and I wanted students to be able to explore and communicate design ideas. Originally the course was more of a general survey including office applications along with 2D and 3D graphics programs. While it is important that they see the spectrum of possibilities and are exposed to professional practice, it is even more important that they gain enough skill in at least one medium to develop their ideas. So we put the emphasis on Autodessys FormZ 3D modeling and rendering (6 weeks) with supporting work in Adobe Photoshop imaging and Adobe GoLive web authoring. As in foreign language immersion, the course begins with a quick introduction to the basics and then includes exercises of increasing difficulty. (Cheng 1997) Students start with web authoring, then learn to create their

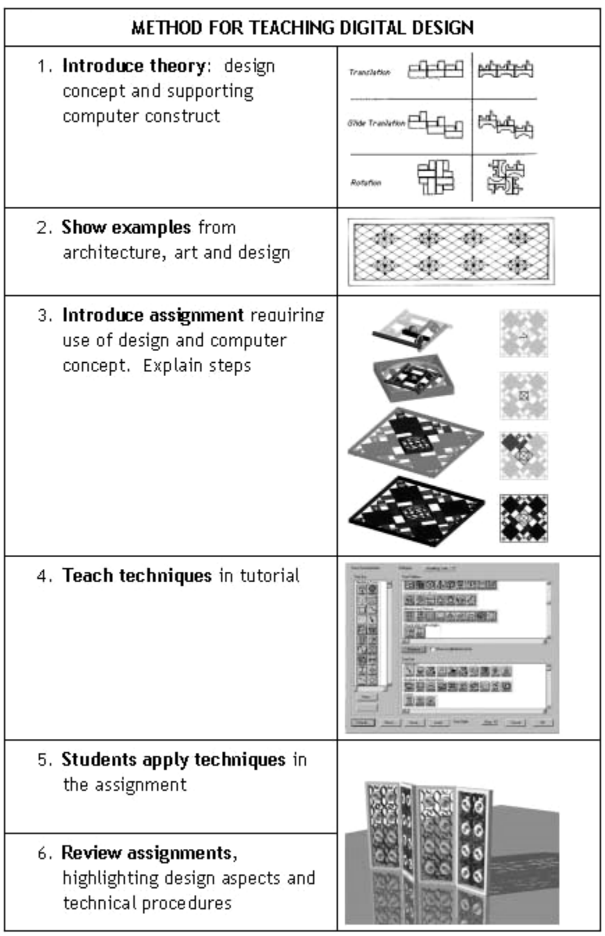

Figure 1. Steps for teaching design concepts with introductory computer graphics.

graphics for their web pages and finally learn how to create 3D renderings that they post-process and post to their web pages. In comparison to earlier teaching of separate applications we now emphasize how to use the tools together.

We have chosen software primarily according to what concepts it can demonstrate, with consideration of cross-platform compatibility and our local context (installed user base, application cost-sharing with other courses, use in the profession, etc.) In balancing robust functionality vs. ease of use, we have leaned towards more robust, complex applications so students could grow into them. In the introductory course, we emphasize the basic steps so that students are comfortable building simple forms. At the same time, there are many other possibilities for the ambitious to explore. Since students are entering with increasing computer exposure, the choice of more robust software is appropriate.

In crafting together a set of assignments for the term, we include a range of project types so that different learning styles are accommodated and successful strategies are exposed. We include both generative and descriptive projects so that students can freely compose elements and later attempt the rigor of careful measurement. As appropriate to elementary skill levels, the projects include some non-architectural basic design: abstract 2D and 3D compositions and transformations that take full advantage of the computer's geometric facility. The first architectural constructions are strongly enhanced by giving students 3D symbols that snap to a preset grid. With this instructor-supplied kit of parts, we can highly constrain the design possibilities so that success is very probable. This follows approaches shown by Ataman (1996), Zhang (1996) and Woodbury (2001) in giving beginners limited elements and operations. Our student work looks more sophisticated because the students start with expertly modeled pieces that give their modeling abilities a large boost.

After using a given kit, students learn how to create and exchange their own components over the Web. The interaction forces students to look at each others work and makes the large class an advantage. (Cheng 1999, Dave 1998) Finally, they methodically model a building from abstraction to detail and show it in context. (see Cheng 1999 for more information)

Organization

In addition to defining what we teach, we need to shape how we teach it. The first step is to set up the framework of the course in terms of scheduling and classroom format. Our course has evolved gradually in this respect. In 1996, it comprised two lectures/week with optional help sessions, supple-

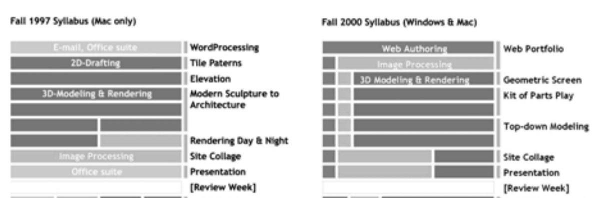

Figure 2. Evolution of applications and exercises from 1997-2000.

mented by Web-based assignments. In 1997, the two 80 minute lectures were supplemented with 80 minute tutorial sessions. In response to student input, in 2000 we changed the format to a single 110 minute lecture a week with a 110 minute tutorial. The greater proportion of hands-on lab time allows students to develop skills more in depth and receive more personal attention.

Up to 1999, we co-taught both incoming freshmen and first year graduate students in the same lecture, with separate tutorial sections. While the format was efficient, the two audiences were receptive to different kinds of approaches. The graduate students appreciated a stronger theoretical basis, while the undergraduates wanted more pragmatic how-to training. In 2000, at the expense of efficiency, we split the class: one class with graduate students in the fall and the other with undergraduates in the spring.

The lab sessions containing 11 to 20 students are taught by paid graduate teaching fellows helped by volunteer teaching assistants. These tutors play a crucial role in the success of the class because many students experience frustration in trying to master complex computer applications. Tutors are available in the departmental computer lab for one hour a week for help outside of class. To coordinate the lab sessions and lectures, the teaching team meets once a week as a group. The professors periodically visit the tutorial sessions to critique the homework and check-in on how the students are progressing. The visits give faculty direct contact with students and allow them to monitor and coach the tutors.

Even with the occasional visits, giving and getting feedback in a large class can be a challenge. Within our computer teaching lab environments, it is difficult to stimulate a discussion about the work. Using data projection to project student work creates an artificial formality and rows of large computer monitors curtail eye-contact and interaction. Hardcopy pin-ups are more successful because they allow easy visual comparisons and they require students to get up and cluster together in a setting more conducive to conversation. Because of pressures to minimize time, money and materials for color hardcopies, we have defaulted to primarily on-line Web-based hand-ins. To supplement the web pages, we require students to create a poster. This helps them become acquainted with the output procedures at the school and allows us to critique their graphic composition skills.

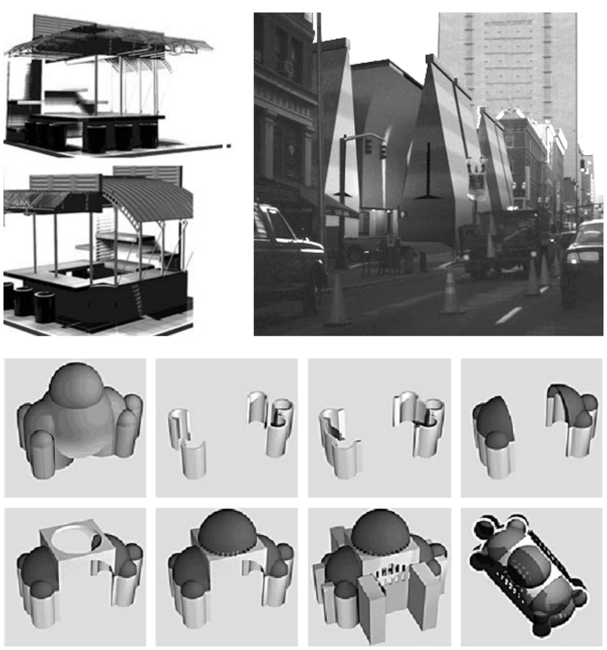

Figure 3. Exercises include kit of parts composition, rendering of model in a site context and case study with Boolean operations. Work by Leit Downing (left top), Josh Brandt (bottom) and Tyler Krehlik (right top)

In order to improve the course, we solicit student input both informally and formally. Students are invited to provide feedback on the course via e-mail and through a simple mid-term evaluation of teaching. At about the fifth week of the class, each student is asked what about the course should be stopped, continued and or started. At the end of the course, students fill out standard university teaching evaluations that are both qualitative and quantitative. While many students voice frustration with aspects that are difficult to change due to budget or manpower, others come up with suggestions that stimulate a dialogue and can be integrated into the class. For example, a survey of undergraduate students in their first year show that the balance between the amount of web authoring desired and taught improved from 1998-99-00.

Tools

Internet Tools

In order to address student concerns, different kinds of teaching tools have been tried with mixed success. Web resources were introduced early by my predecessor and are greatly valued by the students. The course website (http://darkwing.uoregon.edu/~graphics) includes lecture notes, assignments, web and print resources. Online tutorials and a student gallery were added in '98, with the gallery compiled and polished in the summer following the course.

In order to improve communication in the large group, an online bulletin board (MOTET, a product similar to Web Crossing and Web Board) was tried in 1997. It was discontinued after two years because we found it a poor fit and its interface was not tuned to multi-section classes. While we envisioned it as a way to give community to new freshmen who lacked a studio home base, we found that grad students used it more effectively for dialogue. Since most of these grad students shared all classes including studio, they found the virtual community an unnecessary overload.

More successful was the use of Blackboard's Courseware system (http://www.blackboard.com) to supplement the course

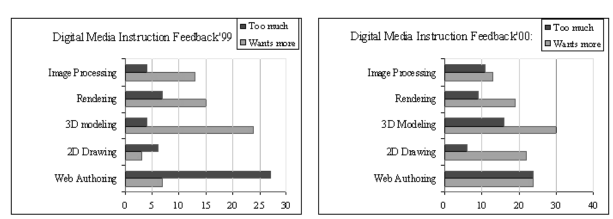

Figure 4. End of year survey of beginning undergraduates on the amount of digital instruction by topic

website. Its streamlined interface allows easy administration of online quizzes, allowing reading comprehension to be tested outside of class time. Students could take the timed quizzes more than one time and follow their quiz and homework evaluations in an online grade book. Predictably, the flexibility of online quizzes proved to be much more popular than in-class quizzes. On the other hand, it would be easy for students to take the quiz at adjacent computers and share answers. From the first year's results it is unclear whether there was any cheating.

In-class Tools for engagement
For classroom teaching, a mixture of traditional and high-tech methods have been used. Web-based or PowerPoint presentations are used to introduce material to the students supplemented by software demonstrations. As a way to make demonstrations more foolproof, we have begun to prepare QuickTime videos that walk through how commands are used in a design context. These allow a procedure to be flawlessly repeated both in the lecture room and at the student's own workstation. Even with improved demos, students find it frustrating to sit through demonstrations without being able to immediately try the commands. For Fall 2002, we will be using a classroom with 20 wireless laptops for small-group activities.

These laptops will update the small-group problem-solving that we have used to supplement lectures. Students are given a question to address in small groups and after time to prepare, students contribute to a discussion of answers and relevant factors. The results can be quickly summarized with freshly uploaded examples or by using a document camera or overhead projector. Spontaneous hand-drawn sketches can be a refreshing counterpoint to prepared static images.

To enliven the lectures and make them more relevant, we draw from the teaching assistants. During the term, each tutor is responsible for briefly showing one of his or her own design studio projects and explaining how relevant digital techniques were employed. The students can more easily understand the reason to learn the software when they see strong examples by their peers. While we show professional examples particularly to illustrate intelligent uses of simple techniques, sometimes the level of polish and sophistication make them seem annoyingly unreachable. Work presented by fellow students is more accessible.

Progress and Challenges
In reflecting on what has been achieved through this dynamic approach of adjustment, the major successes include the development of successful assignments and resources. We have analyzed what students learn from the different exercise types and we use the characteristics to create a balanced course. The problems are clustered in thematic sets (kit of parts, top down, rendering a place) so that a few assignments build upon each other and then are followed by a new set that allows for a fresh start.

We've added practical competency exams as a time-constrained counterpoint to portfolios. The exam requires the students to model and render an architectural element of their own design in a limited amount of time. It gives students a reason to develop efficiency in using the tools and a deadline for refining skills, so almost all students can pass the exam. The exams have revealed that some students have stronger technical abilities than their portfolios showed. These students had weak portfolios because they lacked design judgment rather than technical control.

While some progress is being made, we still work to address the wide range of skill levels in the class. The most naïve beginners have a great challenge and those that can persevere do excellent work, but those who get discouraged tend to stop working. This results in a sink or swim situation: lots of A's & F's especially among the freshmen with poorly developed study skills. We have resisted formally tracking students into slower and faster groups so far, because of an interest in peer learning. Some natural ability clusters appear as we encourage beginners to buy Macintosh computers and teach them separately from those on the Windows platform. We hope to see more peer learning in Spring 2001 by teaching first year undergraduates who are already settled into design studio groups. In the studio they will have a natural support group for their work. While we are encouraged by the results of having a better faculty to student ratio, it comes at a serious cost of efficiency.

With more support for course preparation time, we could put more substantial material online and on video to provide flexible self-study. Unfortunately, the quickly changing software makes it a losing battle to prepare tutorials as they quickly become out of date. It is more pragmatic to rely on well-prepared manufacturer's training material, although that brings up questions of dependency on standardized material. As an alternative, many teachers in ACADIA have dis-

Figure 5. Web page front page by Scott Howe and course gallery page.

Figure 6. Community center design by Padru Kang shows application of digital skills in an intermediate studio.

cussed the need to share course material and develop an online textbook and tutorials appropriate in a wide variety of contexts.

A greater challenge than the temporary nature of course materials is the need to keep equipment updated and working well. Even if technical support is available, the teacher has to be not only the actor on the stage but also the stage manager for the performance. With significant costs associated with the equipment, infrastructure, staffing and maintenance, teacher often have to lobby for funding and work on purchase specifications and personnel hiring. To get the equipment in place and working properly, the person needs to have strong cooperative relationships with computer resource managers and support staff as well as connections to authorities involved in funding. The complexity of these additional duties is well described by the SIGGRAPH Education committee (SIGGRAPH 1995).

External Factors: adjusting the big picture

In addition to the introductory course's content and delivery, the course's curricular context has needed attention. Providing basic computer literacy training prior to the course could even out the entering students' backgrounds. Follow-through practice in studios or media classes help fledgling skills to develop. A logical sequence of classes that interweave technique courses with studio application can make the difference between deep learning and shallow exposure. In considering digital design education to be like foreign language teaching (Cheng 1997), our goal is to immerse the students in a culture of computing. Students need to see how the tools enable the work of peers. They need to be exposed to the different software options available to them and understand that different tools are suitable for different tasks. Finally, having strong role models that demonstrate exemplar work can inspire intensive study.

This paper has described the major challenges in teaching our introductory course along with attempted solutions and partial successes. All aspects of the course are affected by changes in technology: the content, its delivery, staff training and necessary equipment. For all these reasons, teaching digital design is a dynamic challenge. By sharing the lessons we've learned we hope to illuminate roadblocks and streamline the path for others.

References

Ataman, Osman and Lonnman, Bruce (1996) "Introduction to Concept and Form in Architecture: An Experimental Design Studio Using the Digital Media", *Design Computation: Collaboration, Reasoning, Pedagogy*, Proceedings of ACADIA'96, Tucson, pp. 3-9.

Dave, Bharat and Danahy, John (1998) "Virtual Study Abroad and Exchange Studio", in *Digital Design Studios: Do Computers Make a Difference?*, eds. T. Seebohm and S. van Wyk, Proceedings of ACADIA'98, Quebec City, Canada.

Madrazo, Leandro (1998) "Computers and Architectural Design: Going Beyond the Tool", in *Digital Design Studios: Do Computers Make a Difference?*, eds. T. Seebohm and S. van Wyk, Proceedings of ACADIA'98, Quebec City, Canada.

SIGGRAPH Education Committee (1995) "Guidelines for Faculty Teaching in Computer-Based Media in Fine Art and Design", adopted by the College Art Association. October 21, 1995 http://www.collegeart.org/caa/ethics/teaching.html

Cheng, N. (1999) "Playing with Digital Media: an approach to teaching computer graphics", in *Media and Design Process*, eds. O. Ataman and J. Bermudez, Proceedings of ACADIA '99, Salt Lake City.

Cheng, N. (1997) "Teaching CAD with Language Learning Methods" in *Representation and Design*, eds. J.P. Jordan et. al, Proceedings of ACADIA '97, Cincinnati, OH.

Woodbury, R.F., Shannon, S.J. and Radford, A.D. (2001) "Games in Early Design Education. Playing with Metaphor," *CAAD Futures*, Eindhoven, 8-11 July 2001, pp. 201-214.

Zhang, Lei (1996) "The Design of a Test Program for Basic Design", in *CAADRIA '96*, Proceedings of The First Conference on Computer Aided Architectural Design Research in Asia.

***Nancy Yen-wen Cheng, AIA** is an Assistant Professor at the Department of Architecture, University of Oregon.*

REINVENTING THE DESIGN PROCESS. DIGITAL SKETCHING – PLANAR OR ALLPLAN?

Adam Gorczyca, adamgor@arch.pw.edu.pl

"We become, what we have […] We shape our tools , but soon our tools will shape us." (M. Mc Luhan)

The question whether the design process has changed because of CAAD appears to be very urgent and important today. If it is true, then the next question arises - what is the range of these changes and at which phase of design do they appear?

These methodological questions led to a research project on the process of design. In particular the process of forming emergent ideas, transforming them into pictures, and through documentation, to reality. The paper is part of a doctoral thesis, which investigates more thoroughly the influence of CAAD on design methods.

To simplify the scope of this project, only one exemplary CAAD tool will be examined. ALLPLAN FT was chosen because of its usefulness and flexibility. This paper is focused on design methods and their implementation at different stages of design. These stages are chosen arbitrarily from a wide range of "duties of an architect". The criterion chosen was the utilization of a "visualization tool" (which can be both – a pencil and a PC) at the process of creating ideas. Using this criterion, one can distinguish a few stages of design mentioned below:

- **Ideation phase.** It is connected with rapid sketching as well as "brainstorming". Relevant features of CAAD application are: flexibility, easy tool access and fast modifications.