Submissions

Connector is eager to print your experiences in technical teaching and your views on issues of interest to teachers of technology in schools of architecture. It is published by the Department of Architecture at the University of Oregon.

The best way to submit written material for publication is to copy your article, letter, or announcement electronically from your word processing document and paste it into the body of an e-mail message to
connect@darkwing.uoregon.edu

Photographs and drawings are encouraged and may be submitted as prints, slides, or any Photoshop-readable format. Deadlines are February 1 for the spring issue and September 1 for fall. Paper submissions and diskettes should be mailed to:

Connector
Department of Architecture
1206 University of Oregon
Eugene OR 97403-1206

Subscriptions

The newsletter has a circulation of about one thousand readers, and subscriptions to technology teachers in schools of architecture are free. *Connector* is mailed to faculty members teaching technical subjects in member schools of the Association of Collegiate Schools of Architecture (ASCA). If you know of colleagues who should receive *Connector*, please tell them to request subscriptions. Graduate students who teach building technology are also encouraged to subscribe. Requests for subscriptions can be made at:
connect@darkwing.uoregon.edu

or sent by mail to:

Connector
Department of Architecture
1206 University of Oregon
Eugene OR 97403-1206

Structuring Structures

Peter von Buelow, University of Michigan
pvbuelow@umich.edu

In his article in last issue, *Towards Quality Time in the Classroom*, Spiro Pollalis really struck a chord with me. If you missed the article, it's worth the read (or see the more descriptive article: http://www.cdi.gsd.harvard.edu/news/Stretch.pdf). I would like to pick up on a point or two of his, and then explain a bit of what I do in the same vein.

Pollalis makes the point that if your interest is in having the students learn more—in this case, structures—you have to compete for their time. We all have faced it, and it may not even be a bad thing. Students will devote time to studio far in excess of the ratio of credit hours, at the expense of time spent on "support" courses like structures. Under these conditions, the direction that Pollalis describes makes a lot of sense.

The reason students generally have trouble with structures has much more to do with the amount of *time* they devote to mastering the material than to the actual difficulty of the subject. So as Pollalis suggests, anything we can do to help them study more efficiently, or even devote more time to the material, will ultimately improve their understanding.

As obvious as that last statement may be, what do we really do to help students learn? If you give a lecture to a class that is 20% physically missing and 30% mentally missing, how effective is that? You can try to force them to come, but can you force them to stay awake? I agree completely with Pollalis: the better ploy is to give them as many other options to learn the material as possible. Here is where we could really make use of new technologies and media—not just to give glitzy PowerPoint presentations, but to really find ways that will aid students to learn the material.

I've been using PowerPoint this semester for the first time. Though it is a lot of work to set up, the benefits are worth it. It does force you to give thought to the organization of your lecture. That's a plus, and it also allows for easy inclusion of images and audio/video. But perhaps more important, it can be made accessible to the students in a way that a regular lecture can never be. You can post the .ppt version (or a more accessible .pdf version) and allow the students to download it and review it at their convenience. My students have responded very well to having the lectures available in this way, and even pressure me to post them a day in advance, so that they can make print-outs for further annotation in class.

The other suggestions Pollalis makes take some work but I think are worth considering. Videotaping, audio recording, or anything else that might make the material more accessible can't really be a bad thing. And posting it on a web site is even better. Since everyone deals with e-mail, setting up a class forum for Q&A is another good suggestion.

Another element that Pollalis mentioned is breaking the subject matter into week-sized bites of fundamental principles. I attended a lecture this semester by John Holland, the "father" of Genetic Algorithms. He talks a good deal about "building blocks" in relation to problem solving. With GAs the building blocks are binary strings that get manipulated like genes on chromosomes, but he made the analogy to building blocks as the fundamental principles in any field. He made the comment that if you want to solve problems in some field, you must be adept with the building blocks in that field. Adept doesn't just mean you might be able to look it up; it means it is part of the personal tool kit that rattles around in your head and can come to bear when you attack a particular problem. I know the set of building blocks for the field of architecture is a pretty large set, and structures is a subset of that, but I do think we have the obligation to supply the building blocks for structures.

Now for part two of this article: what I've been doing. The component of the structures course that attracted my interest a few years ago is how one handles homework or practice assignments. Solving practice problems is a time-tested method used to learn the subject. You know the old saying: you learn it best when you teach someone else.

If you do it yourself, I guess that is at least second best. But if all you do is just watch someone else do it, but never try it yourself . . .

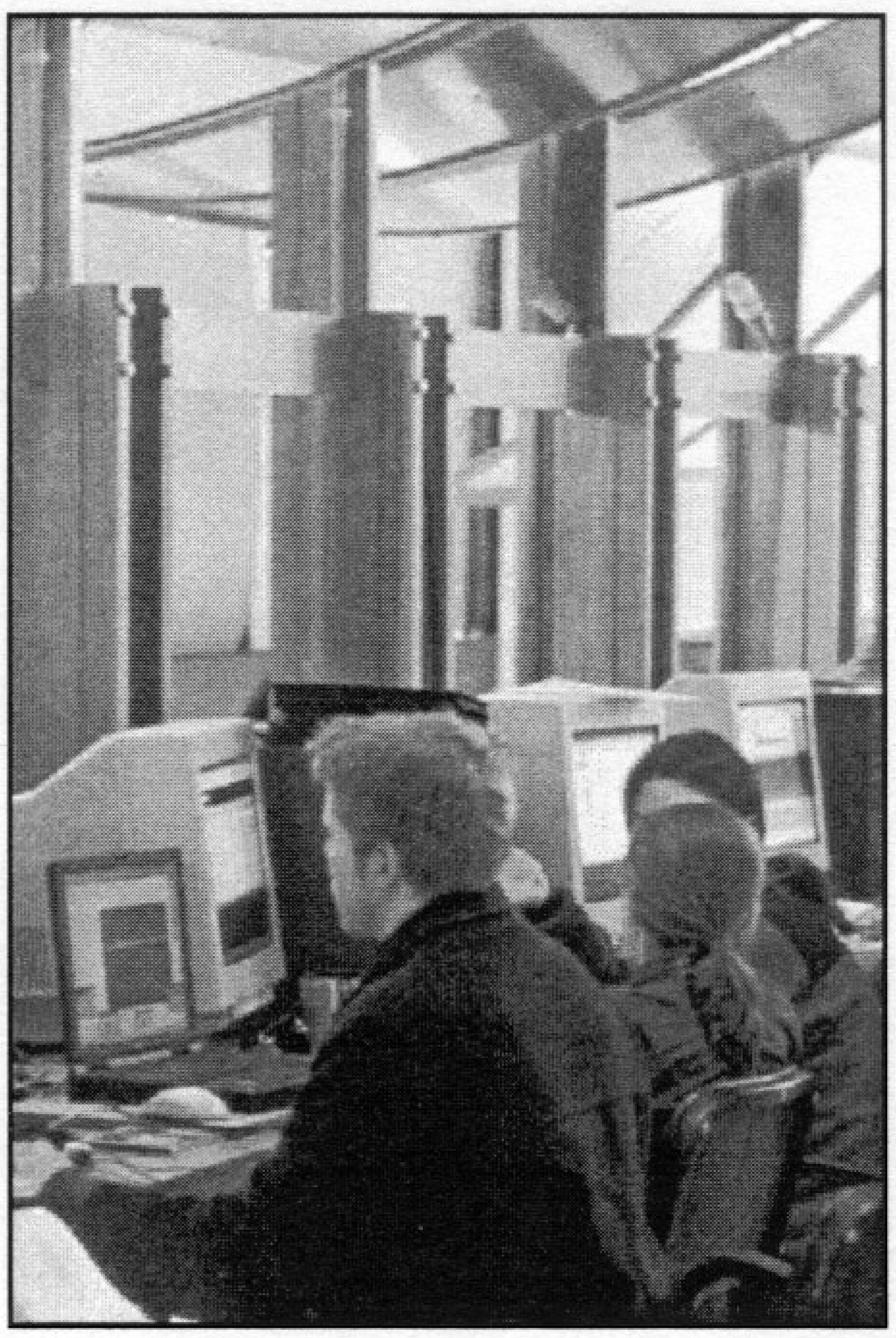
The problem I found was that many students were really not working the problems themselves. From the extreme of photocopied homework to cases of "passive" copying when working in groups, many students don't take the time to work the assignments even once. Working in study groups is good, but in the end you have to actually *do* the exercise. That is the first point with assignments—each student must actually work through the assignment.

I handle this simply by giving each student a unique problem. I set up exercises that are parametric in nature so that each student can be assigned a unique version. They all work the same problem—for example solving beam reactions—but each student has a unique version with different parameters, such as span, load, load pattern, etc. More complex problems can have plenty of parameters, allowing near endless variations.

This very simple adjustment has encouraged several things that enhance the learning experience. One, it eliminates the option of simply copying answers because each solution is unique. So study groups can really function more productively. As a group they can help each other understand the process rather than just copying numbers. In this scenario, study groups are much more effective, simply because everyone is put in an active mode rather than allowing some to just watch passively.

When I first started assigning structures problems, some students would inevitably come to me with the story of how much they had learned in working through a problem but were frustrated by the fact that they had gotten most of the answers wrong, and if only they could do it again they would be able to demonstrate how well they had learned their lessons. With my new way of assigning each student a unique problem, they can simply be given another chance. So where before it was hard to get most students to work the assignment even once, now many are willing do it twice. I have to believe that they are learning the material better in the process.

For exercises to be most effective as learning tools one has to do the exercise correctly, or at least know what's been done wrong. To work an exercise incorrectly and then find out a week later that it was wrong is not really the best way to learn. Even to have the assignments returned in a couple of days would not be as beneficial as having a guiding hand correct you as you make the mistake. Ideally, the tutor might sit beside the student



to prevent long detours down dead end roads, but there are obvious limitations to that. Correction is most effective if it comes at the time the mistake is first made.

My solution to this problem is to have students submit interim answers online, using a program that both prompts them with the questions and instantly scores and returns the correct solution step-by-step. If students get one step wrong they know right away, and will make the correction before proceeding with the problem. This prevents them from going off in the wrong direction or sets them back on the right path.

To summarize, these two ways of handling structures problems have yielded the following advantages:

- Each student works each problem. They cannot passively copy another student's work because they each have different numbers.
- Students can genuinely benefit from working in groups. They see several different scenarios presented by other students' problems and learn many more "what if" lessons.
- Students are given the chance, if they like, to totally rework a problem with new numbers. This takes some of the pressure off mistakes, and encourages a positive "try and try again" learning atmosphere.
- Students are guided to work the problem correctly. If they miss an interim step of the problem, they are corrected and set back on the right course. This happens as they work the problem—while they are interested in the solution—not a week later when they want to "file and forget."
- Students work harder and with more interest because the goals and rewards (on one level) are very immediate and clear.

Overall, I think students learn more in the process. It also relieves me from the burden of grading papers and allows me to devote that time to responding to students' questions.

As the Steppenwolf was told in that enigmatic ticket, these methods are "not for everyone." There is a "price of admission" in the form of a fair amount of programming to make it all work. The parametric problems are the easy part; encrypting, passing, decrypting, storing, compiling individual responses from any machine on the network to a secure location is another thing. But, nonetheless, I think it has been successful and is something that I will continue to develop in the future. If anyone else has tried something similar, I would be interested to hear about their experience.