TOWARD A COMPUTER INTEGRATED DESIGN STUDIO

Bruce Haglund  
Department of Architecture  
University of Idaho  
Moscow, Idaho 83843

Brian Sumption  
Department of Architecture  
University of Idaho  
Moscow, Idaho 83843

ABSTRACT

The formation of our vision for a computer-integrated design studio is outlined. The ways in which our experience in teaching with computers in a variety of settings and in developing our own computer tools has contributed to this is explained. The next step in actualization of our vision is the creation of a design curriculum and a computerized studio which support the integration of this new technology into the traditions of architectural education.

INTRODUCTION

Over the past three years we’ve developed a vision of design education that includes computers as a major resource. We believe that it is important that the role of the computer be integral to the experience of learning to design, rather than acting as a catalyst to the creation of another faction. It is not desirable to have students thinking of design as one thing, technology as another, and computers as yet another. In fact, we see the potential of computers for healing the existing schism between the art and technology camps in design.

COMPUTER COURSEWORK

In seeking to establish a fully computer integrated design program that would live up to our vision, strategies for implementation were developed that would support pedagogical and curricular philosophies while overcoming the problems of limited resources. Initial efforts were focused on the development of these resources. Starting with a single PC in 1984, enough support was generated through the great effort of a number of individuals to allow us to open a microcomputer lab in the Spring of 1985.

Two basic courses relating to computer applications in architecture and computer-aided design were begun at the same time. These offerings provided the spark that ignited our most valuable resource—the students. Most design instructors encouraged students who had taken these courses to explore the use of their new skills in the studio. While these efforts met with mixed results, the influence of the use of these tools was remarkably widespread over a very short time period, probably as much a testament to student imagination and courage as anything else. Enrollment demand for these basic courses grew very rapidly, drawing additional administrative attention and support.

291
Because the vast majority of our students at all levels had little or no exposure to computers prior to these courses, it became necessary to provide first a foundation in the general principles of computer technology and use. This was followed by an introduction to computer graphics and computer-aided design concepts and the development of tool-using skills. References to historical and current influences of tools on design and design process and the implications of future technologies are used to provide an understanding of the context of these new tools with respect to the processes of design.

While both of these courses could be criticized for their overuse and perhaps misuse of commercially marketed software, the advantages of this software use far outweigh the disadvantages if some judgement is exercised. Clearly a CAD package designed for drafting and production work forced into use as a conceptual design tool is potentially dangerous. However, given limited development resources. They commercial software provides an interim means to initiate a program and begin to generate a dialog between students and faculty.

One of our key strategies was to expose as many students as possible to the potential these tools present while developing a broad range of basic user skills. Generating interest by the majority of students, including the better "designers" was of more importance than developing a small circle of "computer" types. By establishing this broad base we are now getting sufficient numbers of students with experience and specific interests to offer second and third tier courses. These offerings include courses in advanced CAD, CAD management, other media, animation, special topics, directed studies, and two in Geographic Information Systems. It is at this stage that encouragement is given to students to develop programming skills if they show any interest in the process of tool-making.

During the first two years of the program, several faculty members, as well as graduate students, were working on specific software development projects, but our major focus of attention remained core literacy and design integration. In the Fall of 1986 we began using the computer in our freshman basic design studio. One hundred and fifty students participate in a series of exercises designed to explore two and three dimensional graphic rule structures combining traditional and computer tools and media.

COMPUTER PROJECTS FOR LECTURES & SEMINARS

Since the initiation of the effort to bring micro computer-based tools into architectural education at our school, we recognized the high potential of using computers for exercises related to technical subjects. The ideal situation would see the curriculum painlessly absorb software packages that were completely user-friendly, had high academic content and contributed to professional training. A few of our experiences illustrate how reality differs from the ideal.

LECTURE CLASS - The protocol used for a large lecture class in Environmental Control Systems (ECS) was to:

1. assign a task that implicitly or explicitly involved a computer tool;

292
2. give a demonstration session showing how the tool is used;
3. and await results in the form of write-ups or presentations.
This protocol, with its brief instruction segment, is dependent on
students' computer literacy and the user-friendliness of the software.
It also assumes that there is implicit academic and professional train-
ing value in the use of the software.

Passive Solar Design was implicated in a project involving determina-
tion of solar heating performance. The software proved to fail on the
criteria of academic content and professional training value. Although
computer literacy was high among the fourth year students who used the
software and the software was professed to be user-friendly, students
were unable to use it effectively because the software acted as if the
computer were a "magic box." Input was a long, complex procedure
involving many screens. When it was complete, the "magic box" would
 crunch the numbers and provide a series of reports. This procedure
not only obscured the connection between the choice of building compo-
nents and performance but did little to relate the act of choosing
performance options to the design process. Moreover, the software was
subsequently removed from the marketplace.

WATT-SUN, which was used for the same exercise, provided a much strong-
er connection between cause and effect because its format allowed the
student to experiment with the input and obtain results rapidly. This
is much more satisfactory academically. However, the software has
limitations in results and scope that detract from its value both
academically and vocationally. The consequence of careless or errone-
ous input results in poor results and obscure error messages, making
it hard to get value from the program when the user is naive or unlucky.
Also, the software limits buildings considered to small buildings in
the state of Washington and other miscellaneous idiosyncratic limits.
These limits detract from its vocational value.

In general, efforts to use software with lecture classes have ranged
from fairly unsuccessful to fairly successful. Moreover, the use of
software in this manner tends to separate it conceptually in the
student's mind from design activities.

IN SEMINARS - The protocol for software use in a seminar in advanced
daylighting was similar in structure to the lecture class protocol,
but different in application. It can be generalized into four steps:
1) Assign the exploration of a daylighting technique that
   involves using software.
2) Give a demonstration of the software's use and capabilities.
3) Consult with students on their efforts.
4) Critique final products.
Again, user-friendliness of the software is vital to the success of
the protocol. The iterative step of consultation and the longer
time frame required for this process-oriented application allows for
more student training time and for more connection to the act of
design.

One piece of software used, MICROLITE, was particularly attractive
because of its simplicity and the visibility of cause and effect in
its operation. Another desirable feature is its malleable procedure
for graphic output of results. Yet again the software proved to
underachieve because of severe limitations in applicability and software flaws. While the software could give impressive results for small rectilinear rooms with windows on one or two sides, it could not be used to explore more complex questions that are likely to be posed by design students.

In this setting, even when the software is limited in its effectiveness, the overall academic value is satisfactory because the focus is on process with the software assuming the role of design tool.

DESIGN STUDIO

The protocol for computer use with design studio has varied widely--from something similar to that of a seminar class to voluntary use by individual students. In an example of the former protocol, one studio used AT&T's Targa-based True-Vision system to study facade treatments in a mainstreet redevelopment project. One member of each design team volunteered to act as the Targa expert for this 10-week project. The use of the tool had limited success because of its complexity and the amount of practice required to gain sufficient skill to use it effectively. In spite of the limited success, students were impressed by the software's features and its potential applicability to design in school and in the profession. The other major difficulty limiting its full integration in the project was its out-of-studio nature; the software had to be used in the computer lab, not the studio. Further reinforcing this separation of design and computer tool were the output media of the two processes (boards vs. slides and models). After the normal studio sequence, however, the students produced and delivered slide presentations which included the computer generated images to the community while using their drawings and models as props. Their sense of success was heightened after this public presentation.

DEVELOPING SOFTWARE

Concurrent with the aforementioned attempts to bring computer tools into the coursework, we were involved in a funded project for development of a computer program to calculate passive solar heating in buildings. Our experience with the flawed computer applications and our satisfaction with generic, yet powerful, word processors and spreadsheet programs affected our approach to writing our own software. We wanted to develop a program that was as specific as the applications programs, yet as satisfying as the generic software. To do this we set the following goals:

1) User-friendly input.
2) Usefulness throughout the design process.

User-Friendly Input. We selected an input format that is similar to a spreadsheet so that data for multiple designs can be input at once. In order to avoid a confusing input path, the main data spreadsheet (Figure 1) is limited to one page. The light-bar cursor that indicates the input position on the spreadsheet is controlled by the keypad arrows keys so that the user has complete control over the order of input. The function keys, F1--F10, perform utility tasks specified by a menu that appears at the bottom of the screen. Most notable among these
Utilities is the HELP! function. Each row of the spread sheet has its unique help screen(s). These help screens aid the uninitiated user in selection of appropriate values for the spread sheet—selections from the help screen are automatically inserted in the main spread sheet. This relegation of detailed information to the help screens allows the experienced user to enter data rapidly without help screens and the inexperienced user to enter data accurately.

Usefulness throughout the design process. At the early stages of design, few details are known about a building, so our software allows the user to make educated guesses about a proposed building. The main spread sheet requires only four facts (building load coefficient, location, passive system i.d., and south aperture size) be known or estimated before a calculation can be made. The spread sheet format allows for side-by-side comparison of design possibilities that may vary parametrically, so the designer can experiment with alternatives.

Figure 1. Main spread sheet with all data fields filled in. Function key menu is at bottom of spread sheet.

Figure 2. BLC Help screen with cross hair cursor positioned to estimate BLC for a residence with a basement and standard insulation levels with a floor area of 2000 sq.ft., 20% of the floor area in glazing, and an air infiltration rate of 0.2 ACH.
Furthermore, the BLC, which usually requires a complex calculation, can be estimated based on floor area, insulation level and building type by using the help screen for BLC (Figure 2). These factors make the software a valuable tool at the initial stages of design. The input data is saved in a file that may be edited at any time. This allows the designer to refine the input as more facts are known and to continue to experiment with alternatives. Moreover, calculations are quick and accurate and output is available in both graphic card numeric tabular forms (Figure 3). These facilities make the software useful as the design process proceeds.

The combination of user-friendliness with usefulness throughout design elevates the software to the status of designer-friendly—user-friendly to a specific audience, building designers.

**Figure 3.** Three output options -- pie charts, bar graphs and tabular.
Table 3 continued.

### CURRICULUM AND PHYSICAL FRAMEWORK

In order to expand our explorations into the development of a methodology for teaching upper level architectural design using these new tools, a pedagogical framework was first established in order to provide an underlying structure to guide us. Fundamental to this structure is the belief that the design studio resides at the heart of architectural education; that it is the environment most conducive to knowledge, growth and ideation; that it is both rich in tradition and opportunity allowing for the exploration of new methods and tools; that it is the focal point for bringing together design data, knowledge and program; and that it is the place most likely to generate dialogue between self, peer and faculty.

Feeling that careful judgement should be exercised when introducing any new system into this environment, we established a series of guidelines to help us develop and/or evaluate systems so that they could constructively co-exist with traditional tools and processes.

As we briefly mentioned, one of the problems with the use of many of the software packages that exist today lies in the fact that they were developed for very specific markets. Product features and structure are often established to meet the demands of a particular segment of the design process, for a discipline other than architecture and are oriented for use in the professional environment rather than education. Software should be sensitive to providing (and requesting) information at a content and representational level paralleling the evolutionary and cyclic nature of the design process. Most analytical programs tend to generate information suggesting a level of significance far exceeding the accuracy of input data available at the preliminary design phase. Drafting packages, when used early in the design process, tend to illustrate design information using an inappropriate level of representation traditionally associated with great levels of investment and imply less freedom or opportunity for change. This
tends to generate an atmosphere of attachment and non-flexibility which is clearly counterproductive to the intent of the exercise.

If new tools are to effectively co-exist in the design environment, then the level and interactivity must go beyond the development of interfaces designed to facilitate the flow of information between software modules from design conception to completion. It must go beyond the development of interactive user-dialogs to include a sensitivity to traditional design media. Designers must feel free to move painlessly between paper-based sketches, drawings, models and electronically generated representations.

While many of these concerns need to be addressed primarily through hardware specifications and software development, the full potential of computer tools used as a cohesive part of the design process will not be realized until systems become physically integrated into the studio workspace. Ideally this would likely happen in each existing studio. Logistically this will not be practical for some time. Yet we cannot expect students to move freely in thought from model to sketch to computer to critique if they must physically relocate themselves every time they wish to shift a train of thought to a different tool or media.

In an effort to address this issue, we have just completed the design phase for a new design studio (Figure 4). This facility, to be completed sometime in the Fall, will allow us to explore issues surrounding the integration of computers into the studio. The program called

Figure 4. Design studio configuration.
for a sixteen-station studio, each station designed for use with all traditional design tools and with the integration of a computer system. Each workspace (Figure 5) includes an eight foot by six foot "L" shaped surface free of any equipment except for a mouse, a digitizer or another input device. CPUs are located under the top surface and monitors are attached to swinging arms above the work surface. This means that the work surface can be used for traditional drafting, sketching or model making.

Figure 5. Eight foot by six foot "L" shaped workspaces.

Because we will need to hold some lecture/lab classes in the space, some compromises were agreed upon in order to allow for non-studio classes of up to 32 students. Work stations are clustered into four unit configurations to facilitate group projects composed of 2, 4, or 8 team members. A crit space is included in the studio as well as an area for model building and large scale plotting. It is anticipated that we will be able to use the facility in a number of configurations, ranging from the rotation of whole studio sections in and out for weeks at a time to clustered assignment of workstations to several different studios working on different projects simultaneously.

CONCLUSION

Our experience has led us to the brink of integration. What happens next should be exciting. We hope that the insights we’ve gained will help us create a studio environment where the art and technology of architecture are brought together.