26
Integrating Computing into an Architectural Undergraduate Program

C. William Fox

Division of Architecture
Temple University

This paper will discuss the process of integrating computing into the undergraduate architectural program at Temple University. It will address the selection and use of hardware and software consistent with the issues and concerns of introducing a new tool to expand the repertoire of skills available to students for use in the design process.

History of Computing in the College

The Department of Architecture, along with the College of Engineering at Temple University, was founded in 1969. Prior to that, the college was a two-year technical institute. In 1978 accreditation was granted for a professional Bachelor of Architecture degree. In 1986 Computer Information Science was moved from the Business College into the College of Engineering and Architecture, and the name was changed to the College of Engineering, Computer Science and Architecture, CECSA. The college presently has four divisions, including the Division of Architecture. The division has approximately 350 full-time students and graduates about fifty students each year with a Bachelor of Architecture degree.

In 1982 a committee of college faculty was formed to research and select appropriate computer hardware and software for teaching computer graphics to engineering and architecture students. At this point, few in the college had any experience in computer graphics. The only required computer course involved very basic programming for word processing and spreadsheet applications. The committee sought to select a system with sufficient capacity for engineering analysis and 3D viewing capability. Choosing the appropriate hardware, software, and peripherals from a myriad of vendors was confusing and compatibility of items was unclear. In 1983 the committee decided to request proposals from several turnkey vendors. The primary justification for this was sole-source responsibility for installation, operation, support, and training. In the spring of 1983 a decision was made to purchase software, hardware, and peripherals from Autotrol Technology. The initial system included a network of four monochrome Apollo workstations, a dot matrix printer, and an HP plotter.

Training was offered with the purchase of the system, and several of us volunteered. Initially frustration was high, for at this point the software had no menus, only keyboard commands, and still contained some errors. With our inexperience, it was difficult to determine when a problem was the result of incorrect input or deficient software.

One year later, in the spring of 1984, the first course was offered to the architectural students, using Series 5000 software developed by Autotrol Technology. On reflection, this course was quite elementary because of our limited experience with the software. The course focused mainly on learning very basic geometric constructions. With additional experience and more confidence, we were able to see the creative opportunities that computers can bring to the design process. In following semesters, the course content has been revised to focus more on modeling, shading, rendering, and 3D drawing (axonometric and perspective). Working with this software is analogous to constructing form with planes, much like a deck of cards or, in conventional terms, like building a model in chipboard.

In 1986, after attending a session at the Graduate School of Design at Harvard, my focus shifted to solids modeling. The opportunity to sculpt, join, and erode blocks to make space offered additional unique alternatives to surface modeling. Inquiry and research led to the selection of I-DEAS, a solid-modeling system developed by Structural Dynamics Research Corp. (SDRC), for automotive and aeronautical engineering design. The college subsequently joined the SDRC University Consortium. In 1987 a new course in solids modeling was offered.

Course Evolution/Development

Students are required to take a computer programming and application course in their second year. The course content and outline are based on the book *The Art of Computer Graphics Programming: A Structured Introduction for Architects and Designers* by Mitchell, Liggett, and Kvan, and the course is intended as an introduction to graphic programming and computing. It is taught by faculty from the Division of Computer Sciences. No additional computing courses are required for graduation; however, two courses, one in surface modeling and one in solids modeling, are offered. The purpose of these courses is to provide students interested in acquiring new tools for design with a theoretical and practical introduction to two alternative methods of graphic computing. These two courses are an introduction to the fundamentals of surface and solids modeling. Both are structured around weekly lectures and supporting exercises. The courses are conducted in a classroom, lecture/lab, and hands-on environment. The lectures and exercises examine the principles and concepts of the respective modeling methods. A final project for each course requires the student to select a small building of architectural merit and make a critical analysis of the project. Either course is required for students desiring to participate in a computing design studio. If a student is interested in acquiring more in-depth programming skills, he/she is encouraged to take courses offered in the Division of Computer Sciences.

Surface Modeling

Arch 190 is generally the first course taken by interested students. Approximately 15 percent of the students in the program take this course. Figures 1 and 2 illustrate representative exercises for learning the fundamentals of generating patterns with lines, geometric constructions, and painted surfaces. In these exercises a simple element is constructed and painted. Generally, these elements are then manipulated into more complex compositions through scaling, rotation, and translation. These exercises build over several sessions and become the vehicle for introducing groups of new commands. Reiterations of these patterns are then examined and scrutinized for visual ramifications in color, texture, and surface characteristics. Initially the exercises are 2D and deal with only the X and Y coordinates; eventually the Z axis is introduced and 3D wire-frame constructions are generated. Figure 3 shows an exploded view of a small church by Schinkel. This wire frame is more complex than an introductory exercise, but it demonstrates the spatial relationships of planes and edges and permits simultaneous viewing of the regulating lines in plan and section. Figures 4 and 5 show examples of a final project. Figure 6 is an example of an abstract design problem which involves manipulating and duplicating two forms to show movement and a hierarchy of spaces.

Solids Modeling

Arch 192, a true solids-modeling course, was first offered in the fall of 1987. Although this course has been offered for only two semesters, the contrast solids modeling offers to surface modeling and the capability of the software have intrigued and interested many students. Figure 7 shows several objects

Figure 1 Polar point construction using translation, and rotation

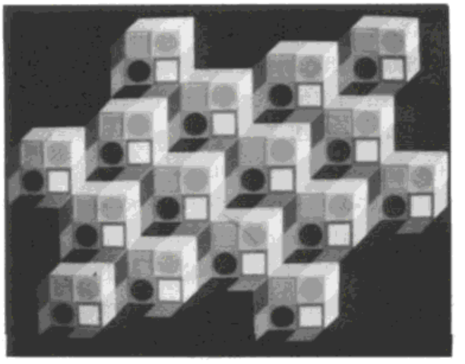

Figure 2 Painting by V. Vasarely, replicated, by using scaling, duplication and translation

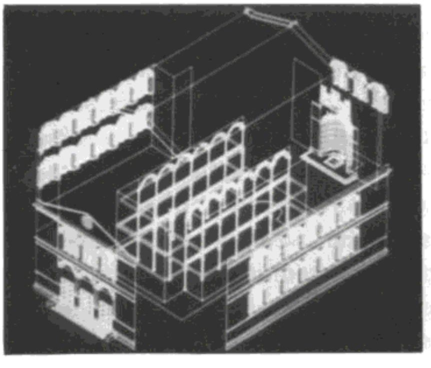

Figure 3 Church by Schinkel, wire-frame exploded view, modeled by D. Gehron

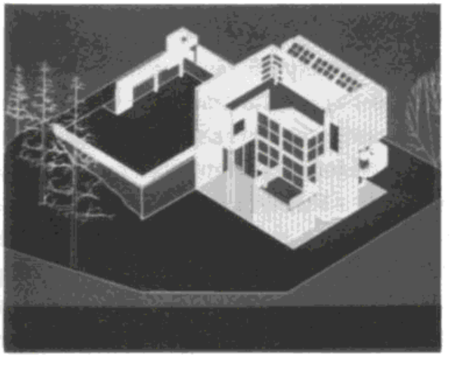

Figure 4 House by R.Meier, painted-surface model by D. Van Horn

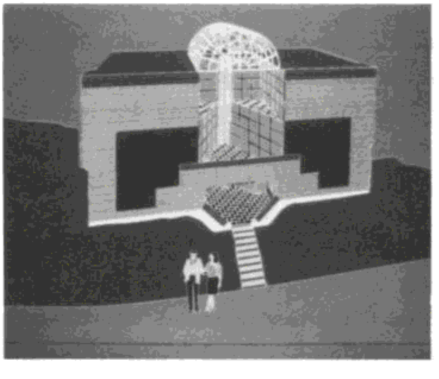

Figure 5 House by M. Botta, painted surface by R. Darus

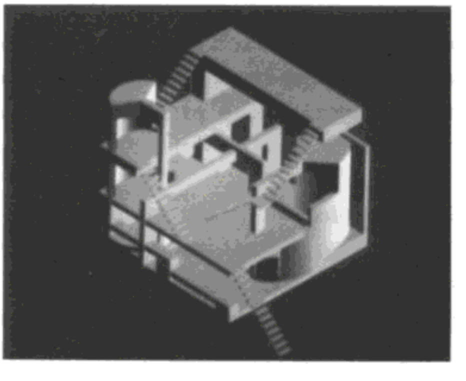

Figure 6 Abstract spatial composition drawn by M. Hollenbach

(primitives) constructed as 3D blocks with the solids software. This is perhaps most similar to building models in clay. Other more complex forms can also be constructed by combining forms with Boolean operations (join, intercept, and cut). This software also allows very sculptural forms to be constructed by making a profile and extruding, revolving, and skinning (stretching a surface over a group of profiles). Figure 8 shows a profile revolved 360 degrees to construct an example of a column used by Frank Lloyd Wright in the Johnson Wax Headquarters building. Figures 7 and 8 are representative of the exercises in this course. Other important options include multiple light sources and surface characteristics, as well as system assembly, and use of a hierarchical data structure. In figures 9 through 14, the individual elements were constructed, and in system assembly instances or clones were created and duplicated into subsystems. In figure 9 the data base subsystems are viewed as a build up or layering of systems, with repetitive elements abstracted. The column shafts in figure 10 were constructed by skinning, and the capitals by revolution. By replacing one element in figure 9 with the more detailed element in figure 10, all the columns in figure 9 could be replaced with the more detailed column. Figures 11 and 12 show examples of how conceptual elements can be constructed, composed, and then later refined, expanded, and exploded in the X, Y, and Z axis into a hierarchical display of the systems. Figures 9 through 12 are all examples of the final projects in the solids modeling course.

General Observations

Most students can learn the fundamentals of surface or solids modeling within one semester. The examples show they can interact with the computer and generate a real architectural building project, inclusive of building components and material detail. New releases of software have become more powerful but still are friendly enough to allow users to learn to manipulate sequences of commands within a semester of work. Generally, the more facile students spend more time at the workstation and tend to experiment with the power and potential of this new tool. Use of the software in these courses is factual and analytical, much like building a model of an existing building in chipboard or clay to study its topology and anatomy.

Computer Design Studio

In the spring semester, a vertical studio, Arch 137/139, is offered to students in the third and fourth year. Within this framework, a design studio, utilizing the computer as the primary tool, is offered for ten to fifteen students. These students must have completed the programming course and at least one of the other computer graphics courses. To date, the computer studio has been offered three times. Each time new areas of computing have been explored and

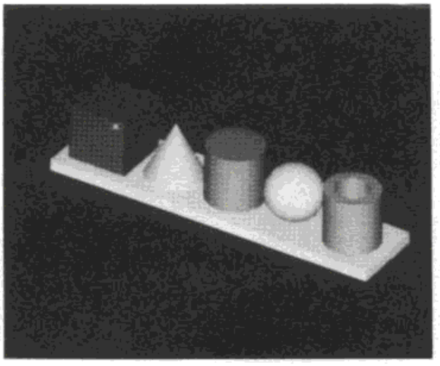

Figure 7 Primitive solid blocks
on plank

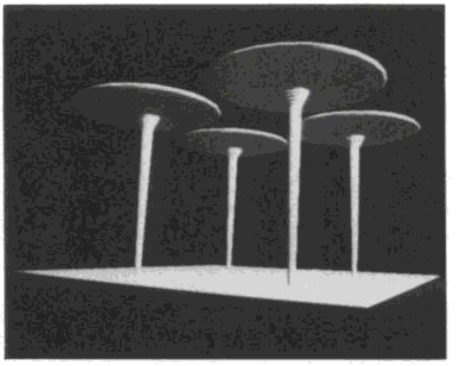

Figure 8 Profile revolved translated resting
360 degrees, columns from Frank Lloyd
Wright`s Johnson Wax Headquarters

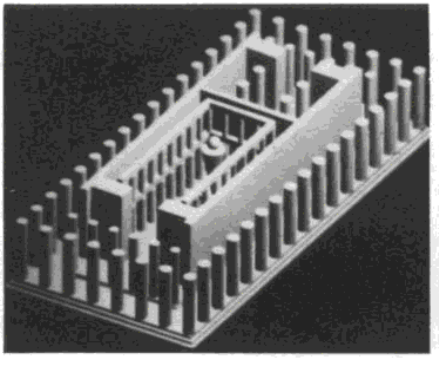

Figure Y Parthenon,
constructed in system assembly
by G. Volk

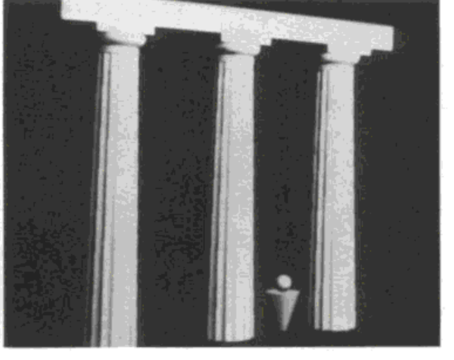

Figure 1(1 Parthenon,
column detail constructed in skinning
by G Volk

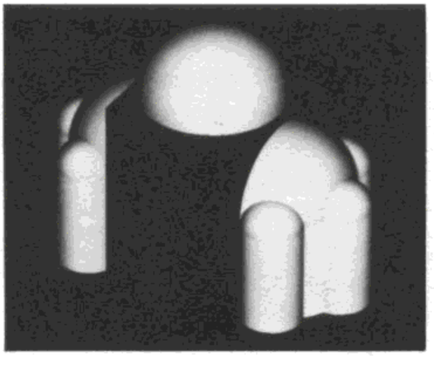

Figure 11 Hagia Sophia,
constructed in system assembly
by R. Darus

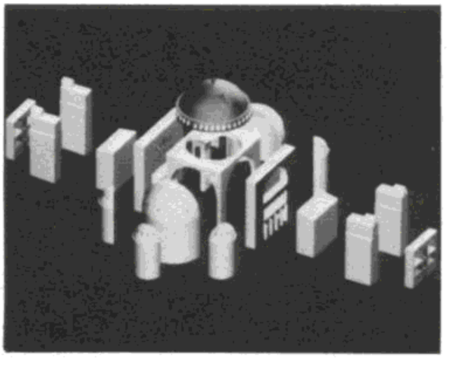

Figure 12 Hagia Sophia,
exploded view, system assembly
by R. Darus

discoveries made. Both the surface modeling and solid modeling software are used. Comparisons are made and examined to study the different influences on the design project.

The studio experiments with alternative ways of using computing in the design process. Although plan and section remain important, 3D views are extremely revealing and helpful in evaluating the form, space, and detail. Reviews tend to focus more on simulating the what and the how of an experience within the space, rather than on the formal relationships of plan and section.

The use of alternative media throughout the semester is encouraged, but emphasis is given to experimenting with computing. Design crits frequently take place in front of the monitor at the workstation. The media used for reviews and presentations include prints, plots, arid slides. Prints and plots are frequently overlaid with soft pencil and / or colored pencil or magic marker. The prints are important for reference work away from the workstation. These sketches serve the same purpose as the conventional yellow sketch paper. Often this information is fed back into the data base and tested as the work is refined. Experimentation of this nature is encouraged to test alternative ways of using the computer as a tool to facilitate a more comprehensive design process. Figures 13 and 14 are views of an abstract spatial composition using two contrasting forms, one rectilinear and one curvilinear. These were then manipulated to form one large space and three smaller spaces. The intent was to take some risks and explore space that would be difficult to explore using conventional tools. Figures 15 through 18 are examples of modeling fragments of existing buildings to show architectural detail.

The shading, light sources, and 3D viewing (perspectives) all allow more extensive evaluations of a design, as the designer can view the form and space as the user will actually experience it. Use of this powerful tool, through the design process, appears to place less emphasis on the formal relationships and more on the evaluation and analysis of the actual spatial experience.

Evaluation of the Past and a Look into the Future at Temple

Looking back over the past five years, I sometimes feel that we have accomplished a great deal, but in that same period of time the software and hardware have changed so rapidly that there is still a great deal to explore. From a more pragmatic point of view, there are some fundamental issues that need to be addressed for computing to be more fully embraced within the Division of Architecture.

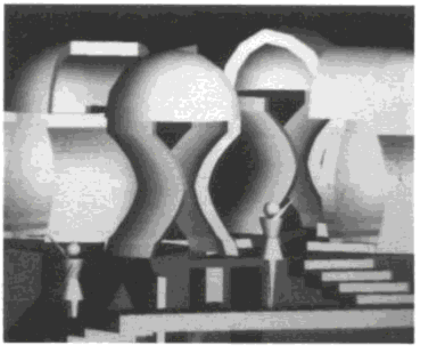

Figure 13 Spatial design problem, design studio, by G. Volk

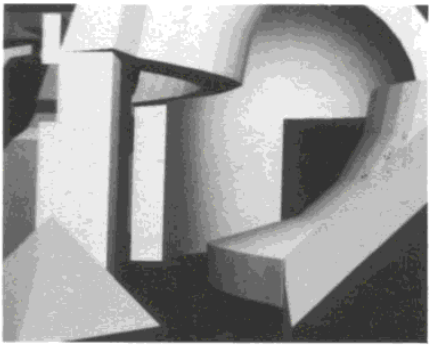

Figure 14 Detail, spatial design, design studio, by G. Volk

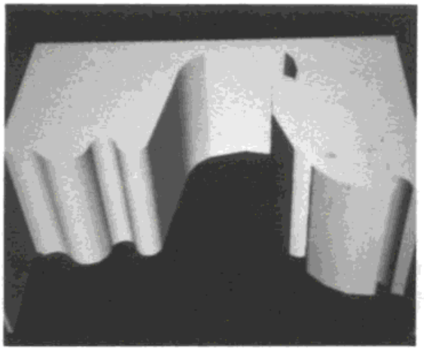

Figure 15 Church by Utzon, model of roof by R. Rozman

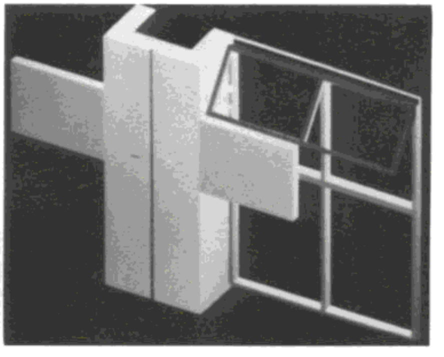

Figure 16 Office building, M / G model of facade repetitive element

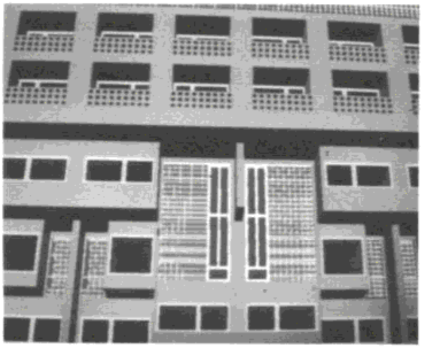

Figure 17 Building by A. Rossi model of facade by G. Volk

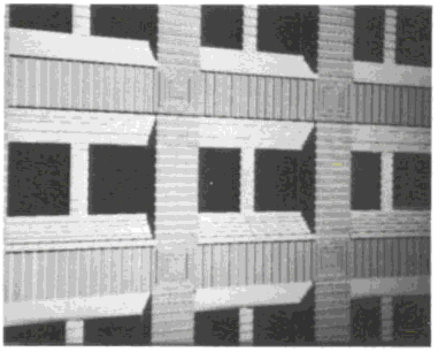

Figure 18 Building by M. Botta model of facade by S. Bwint

The computers are currently in a 'computer lab five floors from the studio. This means a real effort must be made to use the facilities. This set up may work for a graphics course, but is totally inappropriate for encouraging spontaneous studio use. This fall, to ease this condition, we are installing a Macintosh lab adjacent to the studio spaces. This will permit students to move freely between their drafting boards and the workstations. We hope eventually to locate workstations within the studio.

The department has no intention of offering any more than a limited programming course to the architectural students. If a student is interested in programming, courses are available through other departments. The real question is how many different types of hardware platforms and what software applications should be made available, and how they should be introduced in the program.

The computer graphics studio is only in its infancy at Temple. This past semester was the first time related design issues of program, conceptual development, and architectural detail were dealt with in a comprehensive way. The process of design development is still fragmented and sometimes short circuited. Grits are often consumed with the technical process of constructing the database. Key to this is understanding the development of appropriate databases, their size, and the required generation time. Perhaps this is a temporary condition until more powerful workstations are available at affordable prices. In the meantime, alternative methods of utilizing the computer and a more flexible interchange with conventional tools all need to be tried and tested. The power and capability of the tool cannot be denied, and the studio is the best location for integrating its use and exploring its potential. More effort by everyone needs to be applied to the integration and mainstreaming of its use.

File storage is of key concern. As the projects become more ambitious and data bases increase in size, more disk storage is needed. Although we have approximately one gigabyte of storage in the network, this was filled two weeks before semester projects were due. During the last two weeks rigorous management of files was necessary by the systems manager. Lack of adequate storage limited development of projects, and subsequent crashes often involved loss of important files. This could be alleviated with more disk storage space. As the workstations and the software become more powerful, it is inevitable that projects will become more complex and that there will be a greater need for more storage. What is enough storage?

How would we like to expand the computing equipment? Always by more than the budget allows. We currently have fourteen Apollo workstations on two separate networks: the original network, which at this point seems antiquated (five years old), and a new network of eight more powerful Apollo workstations (DN3000s to DN4500s), an ink-jet printer, and a pen plotter. Only these eight are used for studio work. They are in high demand by both architects and engineers. Students must sign up for two to four hour blocks of time, and frequently the lab is open twenty-four hours a day when projects are due. Obviously more workstations would alleviate this condition, and as funds become available workstations will be added. In addition to more workstations, we would like to purchase a scanner for photographs and slides, photographic equipment that captures an image through the workstation and a paint package for touch-up work in image processing.

At Temple through telephone connectivity, all students and faculty can access the library stacks and, with an account number, the Vax mainframe. This coming year the various networks within each college will be connected to the larger university-wide network, allowing files to be shared between many workstations on the campus. This will also allow movement of files between the Macintosh lab in the Division of Architecture and the Apollo lab in the college.

In an article about the scholars workstation of the 1990s in the October 1988 issue of *Academic Computing, R.* Weissman predicted more powerful workstations with up to 100 MIPS, 130+ MB of memory, and a gigabyte of local storage. Predictions also indicate that future networking and file exchange may be comparable to today's Fax communication via the telephone. With this capability, many of the technical concerns I have cited above will disappear. Then, perhaps, we can really explore the creative potential of computers in design.