ARCH 431: COMPUTER-AIDED DESIGN

J. Peter Jordan, AIA
Associate Professor
School of Architecture
University of Hawaii at Manoa
Honolulu, Hawaii 96822

ABSTRACT

There is a significant variance in the way computer courses are taught at various institutions around the country. Generally, it is useful to think of these courses falling either into a "tool-building" or a "tool-using" category. However, within either category, there is a variety of focus on the application of the "tool". Two courses have been developed at the University of Hawaii at Manoa which deal with computer applications. The first course is more quantitatively oriented, encouraging students to explore ways of dealing with problems in a more complex and substantial manner. This paper deals with the second course whose focus has shifted toward design issues, using the computer as a tool to explore these issues. This course exposes the student, not to training on a specific computer-aided drafting system, but to issues in computer-aided design which include hardware and software systems, human-machine interface, and the nature of the design process. This course seems to be an appropriate model for introducing computer-aided design to undergraduates in a professional design program.

INTRODUCTION

Over the past ten years computers have become manifestly useful tools in the professional practice of architecture. Concurrently, there has generally been more reticence to incorporate these tools into the architectural education curriculum. Numerous surveys in the last five years document the growth of computer courses in various schools while others have studied the pressures which are acting on institutions to incorporate computers into their programs. It now seems clear that architectural education must include courses dealing with computer applications if for no other reason than pressure from accreditation agencies.[1] Obstacles to implementation of computer courses have varied from one institution to another, but common problems in the institutional environment include lack of interest and/or expertise among the faculty, lack of administrative support, facility limitations, and budgetary constraints. A more subtle obstacle to the deployment of courses addressing computer applications has been methodological uncertainty.

It is generally useful to think of the various methodological approaches as falling either into a "tool-building" or a "tool-using" category. Neither has emerged as a dominant approach, nor has there seemed to be any general consensus for appropriate implementation among either "tool-using" or "tool-building" constituencies. Moreover, there has been a significant variation of application to which the "tool" is being put. The methodology for introducing computer applications seems to be determined more by environmental parameters than by academic policy. One school may be unknowingly offering computer courses because one instructor has decided it is time and has "bootlegged" a "Turbo-PC" into a particular course or studio. Another department's program may be centered on underpowered PC's whose speed and graphic capabilities are only acceptable when the alternative of their absence is considered. Still another school many have implemented a program based on an instructor's special interest in computer applications to a particular aspect of the curriculum.

ACADIA's conferences have served as forums for representatives from various schools to air their approach for review by their peers. In 1986 and 1987, the range of applications has extended across the curriculum from general design studio applications to environmental controls to history to color theory. This paper shares the methodological approach chosen for the computer-aided design course within the School of Architecture at the University of Hawaii at Manoa (UHM).

COMPUTERS AT THE SCHOOL OF ARCHITECTURE

Since the late 1970's, it has been a requirement for all students in the University of Hawaii's BArch program to take two 3-credit courses in computer applications. The first course (ARCH 235) is an introduction to computers for architectural students and assumes no previous knowledge of computers. It introduces computer basics such as hardware and software and moves to applications of various systems to architectural problems. This fundamental structure tends to emphasize the validity of courses in the curriculum which expose students to quantitative applications.

The more advanced course (ARCH 431) began as a framework where students could explore one of a variety of computer applications in more depth, an interim approach while the School acquired a CAD/D system. During 1985 a system was purchased-- Prime Computer's "AEC Medusa" system installed on a Prime 750 mini-computer with seven graphic terminals and a pen plotter. In January, 1986, after the instructor had received the manufacturer's training, the course began with a group of eager students. The initial bias was toward addressing production issues since that was the instructor's professional context. Indeed, the training given to the instructor by Prime Computer was exclusively focused on using the system to produce contract documents.

COURSE DEVELOPMENT

CAD/D systems were still relatively rare in Honolulu, and since many of the students had not even seen a system, they were excited. The syllabus for the first semester followed the manufacturer's training although there were modifications to the schedule and development of a series of exercises which were slightly more complex than the training exercises from Prime. (Figure 1 shows an exercise in which details were "drafted" on the system.) The students remained eager and enthusiastic even when confronted with an underpowered system and an occasionally confused instructor.

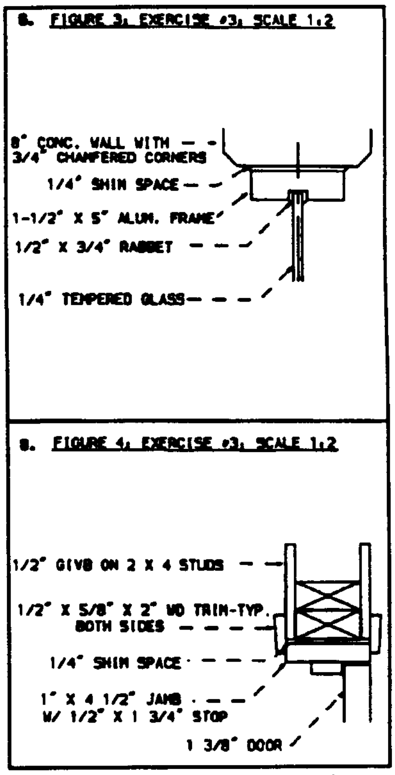

Figure 1 - By Susan Irvine

In assessing the semester, it seemed that the strong orientation toward production issues (several exercises dealt with generating details and floor plans) had been a serious problem for the majority of students who had no previous office experience. Several exercises had become complex because of a desire to make them more "realistic". The final evaluation was that the nature of the course had been little more vocational electronic drafting.

During 1986 the instructor attended the NCGA (National Computer Graphics Association) Conference in Anaheim and ACADIA '86 in Houston, carefully listening to presentations relating to computer-aided design. Faculty and administrators at Texas A&M University, University of Houston, and Rice University were generous in sharing information about their computer courses and the goals of their curricula which related to integrating computer applications. Other resources included several articles related to design language nearly all of which cited Summerson's *The Classical Language of Architecture*[2] as a source. Reviewing relevant selections from the work of Noam Chomsky and his followers on linguistics and grammar led to a better understanding of the possibilities in applying the linguistic metaphor to the concept of design language. The result was a conviction that design and not production issues should be the focus of the course. Many of the ideas presented in the course are based on the work of others; however, the approach and content of this course offers a divergent, if not unique, methodology.

By the beginning of the Spring 1987 semester, the course had been redesigned; new readings, new exercises, and a new conceptual basis for the course. Now in its fourth consecutive semester, the course's fundamental orientation and structure has remained intact; however, numerous modifications have been made. During that Spring, the decision was made to teach this course during both the fall and spring semesters in order to attempt to balance the demand on the computer system. In the Fall 1987 semester, a 3-D solid modeling exercise was introduced which was subsequently modified in the Spring 1988. In Spring 1988 a teaching assistant was appointed to do most of the instruction relating to the actual operation of the computer graphics. The class has been divided into two parts, each generally spending one session a week attending lecture and the other in a computer lab.

ARCH 235, the earlier computer course, is a prerequisite as well as a minimum of four semesters of design studio. Because of the facilities, enrollment is limited to twenty students per semester. One faculty member and one teaching assistant are responsible for the course. There are three explicit goals that students must meet if they are to earn a satisfactory grade (C or better) in the course. Those goals are:

1. Learn how to operate and use the Prime Medusa AEC CAD/D system and apply CAD to design problems.

2. Read literature on integration and application of CAD/D system in the professional office.

3. Respond in writing to various issues raised by the implementation of computer-aided design and drafting systems in the architectural profession.

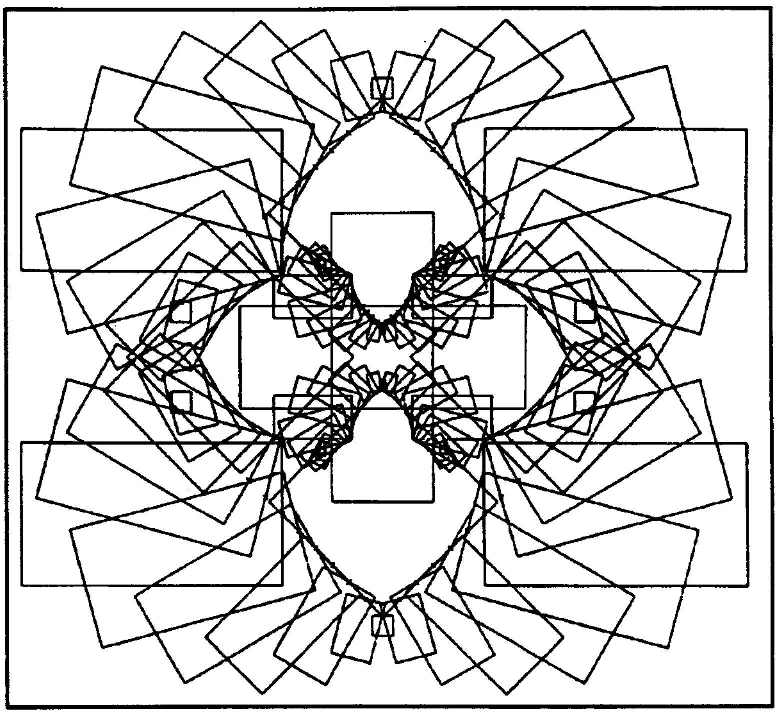

Figure 2 - By Dean Ichiyama

Students are required to master a set of operating system commands as well as graphic commands. All assignments with the exception of the first are distributed on the computer and students must know how to copy the master file into their own directories. Students are also expected to read a number of selections which include design theory, computer-aided design issues, current research and exercises at other universities, and an exposure to current applications in professional design offices. Although there are some "classic" readings, an effort is made to keep most of the readings current. Students are required to write three essays which focus their thinking on issues raised in the course. The key issue with which the course confronts the students is how a computer system can be used in the design process. Students are expected to deal with this issue cognitively in their writing assignments as well as experientially in their computer exercises.

DESIGN THEORY, COMPUTER ASSIGNMENTS AND CRITIQUE

The course begins with a presentation of the primary structure, assumptions, and goals of the course. The students are told that the critical basis is design language; and as such, its design orientation is formal (as in *form*-al). Other design theories may be introduced, but the primary focus will be on form. Anyone who would rather be enrolled in a computer-aided drafting course is directed toward the community colleges where a number of such courses are offered.

The first task is to orient the students to the computer graphic system at UHM. A preliminary exercise designed for students to acquaint themselves with a limited number of operating system commands is initially assigned. The second task is a presentation of the fundamental ideas of design language. Summerson's book is used as a starting point. The students are introduced to a linguistic understanding of language as vocabulary, grammar or syntax, and semantic. Chomsky's nomenclature of phonemes, morphemes, phrases, and paragraphs is introduced and correlated with architectural design. The simple forms (phonemes) are combined, using certain syntactical (or grammatical) rules, into morphemes which in turn are combined, using another set of rules, into phrases, and so on. An analogy is made with a hierarchy of architectural forms starting with the very simple to the complex. Vocabulary (or form) is presented as being a separate component of design language from syntax (proportion, symmetry, scale, rhythm).

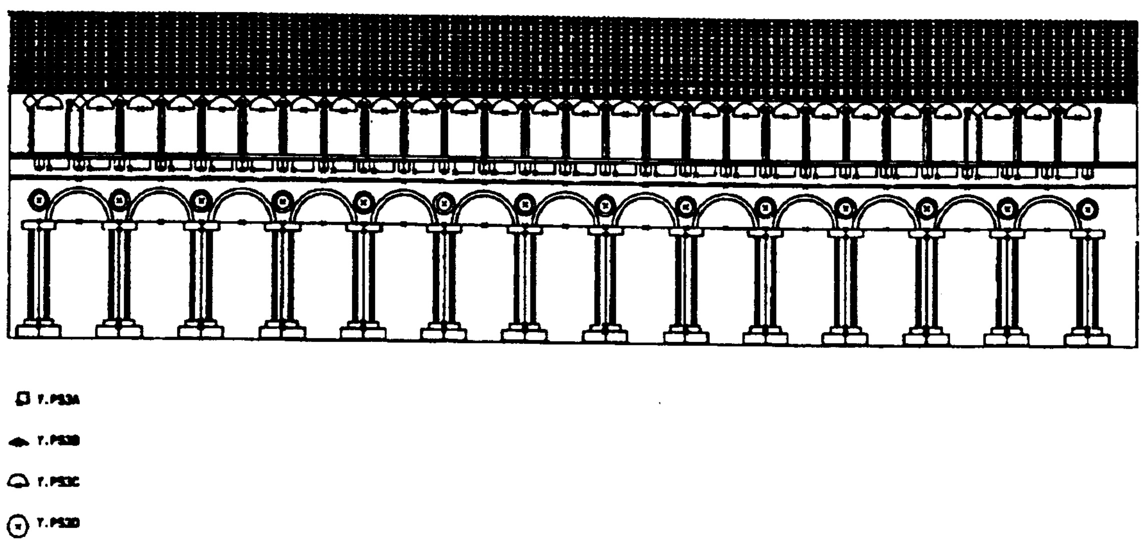

Figure 3 - By C. J. DeLeon

Reference is made to major works of architecture in order to illustrate these points. Slides of architecture from various periods are used as well as reference to various buildings on the campus and in the community. Basic rules such as symmetry, proportion, rhythm, and scale are discussed and related to transformation, rotation, and translation. This gives the class a common experience for the discussion and analysis of course assignments as well as general architecture.

The mechanics of plotting are introduced after the first graphic assignment. Thereafter, all work is required to be plotted, and the results are pinned to the wall and reviewed in a classroom setting. The critique comes on two levels: mechanical and aesthetic. Mechanical issues are dealt with at the beginning of each class session with students asking questions about problems they may have had with producing the assignment. As the semester progresses, less time is required for mechanical problems. The primary focus of the critique is to raise questions about the designs produced. It is in these sessions that design ideas are related to student work. Because this course is not listed as a studio and because the basis of design work is narrowly focused, students seem to be more willing to experiment, producing design work which might be rejected out of hand in a studio session. Students are encouraged to participate in these critiques, challenging the instructor as well as their peers.

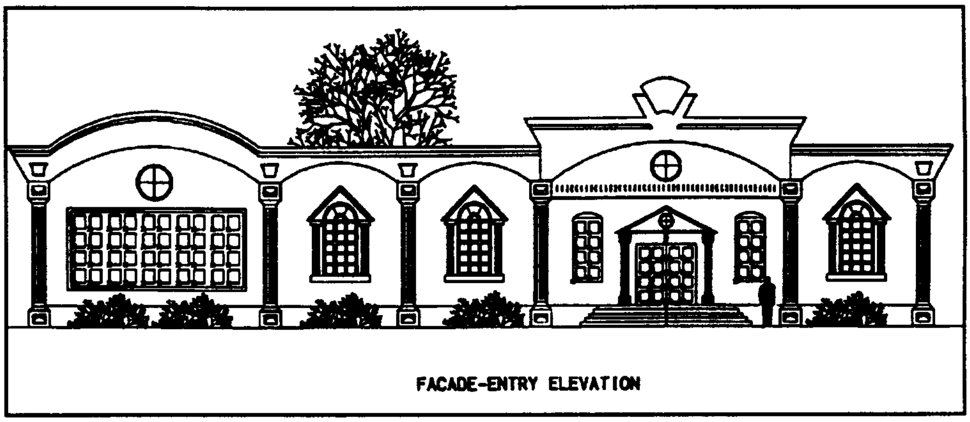

Figure 4 - By Stacey Tangonan

The first two exercises require that the students develop a set of rules for creating various graphic patterns by loading a primitive into the graphic database. They must deal with issues such as scale, rotation, and translation in developing their rules. A simple shape (such as a square) is given as a graphic primitive (called a symbol on this system) for the first exercise. In the second exercise, students will be using one of the patterns created in the first. They may modify their symbol by rotation and magnification, but they may use only a single symbol. Designs tend toward the abstract and graphic although some can be quite architectonic. Students are able to generate designs which can become somewhat complex without knowing very much about how to operate the computer. They tend to be frustrated by their unfamiliarity with the computer, and they often express their discomfort with design process being so rigorously procedural and objective. However, during the second exercise students begin to feel more comfortable with the system during the second exercise, and students begin to become more experimental. Figure 2 shows student work for the second assignment.

Students should have learned to load symbols into their file and how to control their location. They are using move, rotate, and magnify commands, and they have learned how to group elements to create their own symbols which are subsets of a drawing. They are dealing with plotting which necessarily means dealing with scheduling and maintenance issues. They have not yet been exposed to the process for drawing lines.

The next feature of the system that they are introduced to is the ability to "permanently" associate elements into nested hierarchies of elements. Prime's Medusa system identifies four types of elements: lines, text, prims, and clumps. Clumps are groups of elements which may contain lines, prims, or other clumps. Groups, introduced in the second assignment, are temporary associations; and, since they are defined with a line, they tend to be geographic in nature. Moreover, only one group can exist at a time in a given database. In order for a new group to be created, it is necessary to delete any existing group lines and draw a new one. Clumps are logical associations which retain their associativity. Students are taught how to create clumps, how to move through the logical structure of the database, how to perform familiar operations on clumps (move, rotate, etc.), and how to find a particular element (text, line, or clump) in the database. Emphasis is placed on the logical structure of the drawing as opposed to its aesthetic or geographic structure.

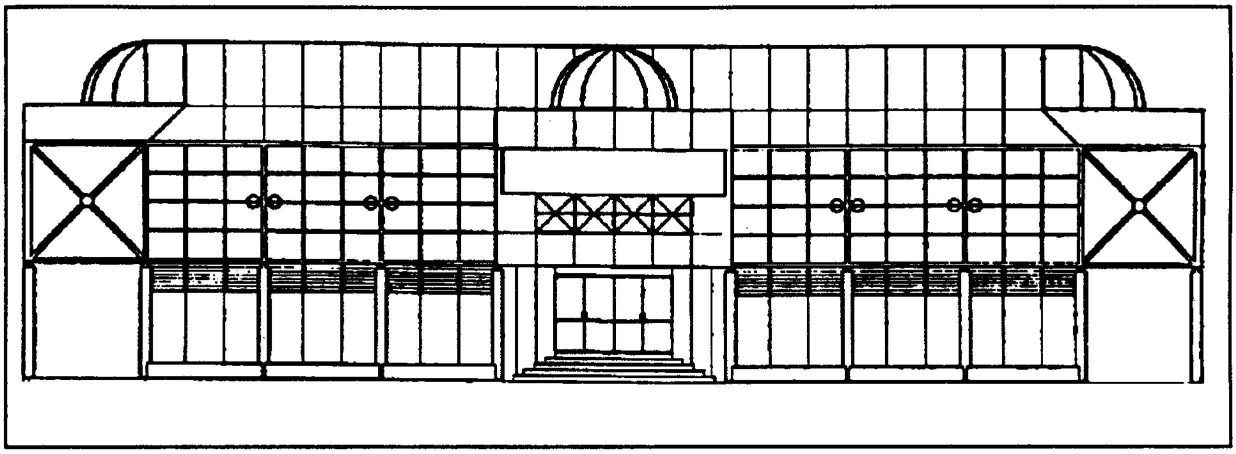

Figure 5 - By Clifford Chua

The third exercise requires the creation of an elevation from a limited number of primitive graphic elements. The graphic elements are four primitives: square, circle, equilateral triangle, and semi-circle. In order to create any level of complexity, students must group elements; and, in order to deal with these groups with any ease, students must group elements in clumps. Students are given a horizontal rectangle created at a certain scale and dimension which forms the base for the elevation.

Results can be awkward; the assignment is considerably more complex and students are uncertain about the concepts and mechanics of clumping. Some work is very classical even down to an attempt to mimic classical vocabulary (see Figure 3). The remainder usually has a modern vocabulary of simple geometric forms with either a symmetrical or asymmetrical grammar. Some work is complex and exhibits a great deal of effort although a satisfactory solution is neither necessarily complex nor simple. Review of the work begins to make the students more aware of what they are doing and what they are trying to do in terms of their design. At this time the semantic of their design is discussed; and it is pointed out that if language is to communicate (its function), then it must consist not only of rules and form, but also of meaning.

The fourth exercise is identical to the third, except that students are allowed to create their own symbols. They have been introduced to the various line operations (straight lines, circles, arcs, etc.) and edits (move point, clip, extend, divide, etc.). The results tend to fall into the same categories. The major differences between work in the third and fourth exercise is the degree of proficiency and diversity. Students are clearly more confident about handling their work on the system. They want to push the limits of their understanding of the system and the design theory. Some are simple; classical elements may be used, but there is little detail. Others may be more ambitious and attempt a contemporary solution within a classical vocabulary (see Figure 4). Others use conventionally modern design forms, but with a very classical syntax (see Figure 5).

This system does not readily handle three dimensional (3-D) work. 3-D modeling takes time; more than six hours for three to four relatively simple models being generated at the same time. Discussions with other users indicate that the processor is underpowered for this application. There is an interactive viewing mode which is acceptably responsive for generating images. However, importing the images to a drawing file for plotting is the task which takes an inordinate amount of time. The product of the modeling cycle is a 2-D representation of the model and is a solid model whose quantitative attributes are more useful for mechanical engineering than architectural design. Because of these limitations, students were not initially introduced to this aspect of the system. However, after two semesters, the decision was made to expose students to this facility despite the disadvantages. A result of the original 3-D assignment is shown in Figure 6. Some routines have been written which allow modeling to be done easily in a batch mode so students are not required to "baby-sit" their workstations while the computer is working. A macro has also been implemented for the interactive viewer which allows students to see their work from a variety of viewpoints without being adept at the viewer commands.

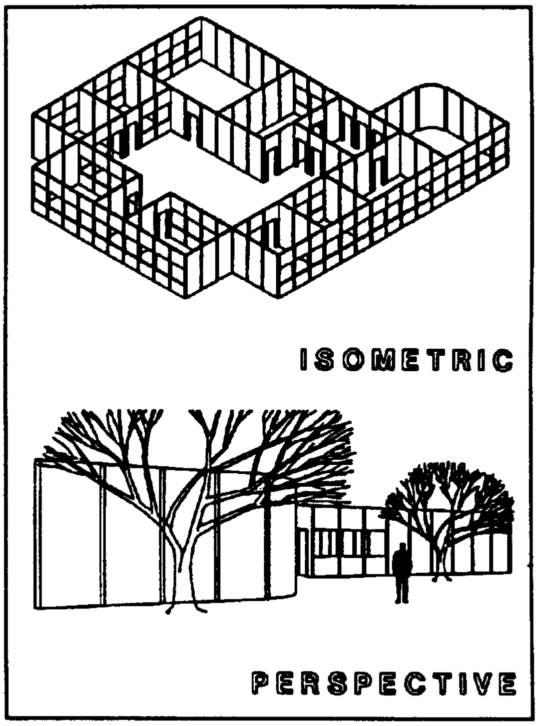

Figure 6 - By Clifford Chua

The current 3-D assignment involves the mass modeling of a high-rise project in a "downtown" setting. Students are given a nine-block area with two empty sites. One is to remain as open space, the other is used for their site. Their response is mixed. They are nearly uniform in their appreciation for the exposure to this tool and are always enthusiastic when the exercise produces the expected set of images (Figure 7). However, this assignment takes an amount of time and patience which is out of scale to the results. Moreover, some of the students are working in offices which are producing more sophisticated visualizations in less time on less expensive equipment. The 3-D module of the system is not consistent with the 2-D environment, and it is therefore difficult to reinforce design theory in this portion of the course. To a great degree, this exercise is non-continuous with the rest of the course, both in the way students interact with the graphics system and in the way design ideas are related to student work.

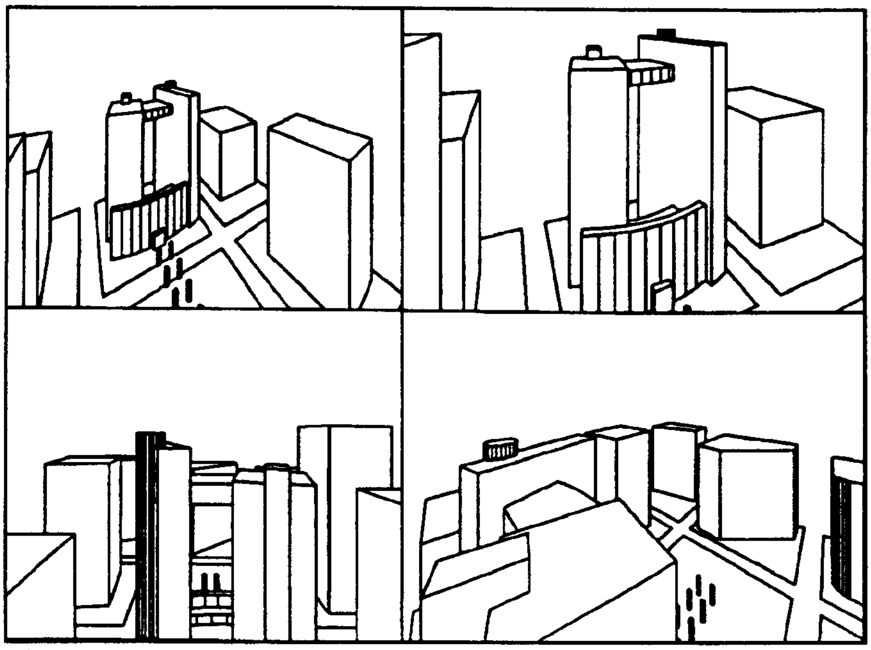
Figure 7 - By Beatrice Ku

The final exercises return to the two-dimensional environment and are accomplished in two- or three-person teams. An introduction to the sixth exercise emphasizes proportion as a design rule. Excerpts on Renaissance and Gothic proportioning systems for describing buildings are read to the class. They are required to choose a building from a selected list and analyze the major elevation on the computer. Most of the buildings can be found on the campus although there are some from off campus. Students are strongly encouraged not to work from record drawings, but from first-hand observation as well as personal sketches and photographs. In this way they may begin to see proportional relationships that they may have missed in a mathematical analysis of dimensions on a drawing. This meets with some skepticism, but the focus of this exercise is to use the computer to "see" the building through a proportioning system rather than a set of dimensions. The final assignment is to present the design language of the building without drawing the complete elevation. This is an exercise not only in analysis, but also in graphic composition. It is suggested that if students use clumps in the analysis and presentation of this exercise, then the final exercise will be much easier.

There always seem to be one or two teams who really understand the challenge of the assignment and submit work which is perceptive and graphically communicative. Some are facile enough with the computer at this point to accomplish some really interesting effects (see Figure 8). On the other hand, there is that one team which seems to miss the whole point; slavishly copying some details from a working drawing and submitting something which appears to be an acceptable analysis, but is graphically unexciting.

For two semesters the final project was to do four elevations of their building. Those teams which really understood their building and managed to organize the vocabulary into clumps were ready to breeze through this final assignment. Those who were not well organized or were too tied to a set of drawings had a good deal of work still ahead of them. The results of this exercise were usually acceptable from a graphic point of view, and many students carry plots of this exercise in their portfolios (see Figure 9). However, their appreciation for the computer's role in this exercise is directly proportional to their understanding and success in dealing with the previous exercise.

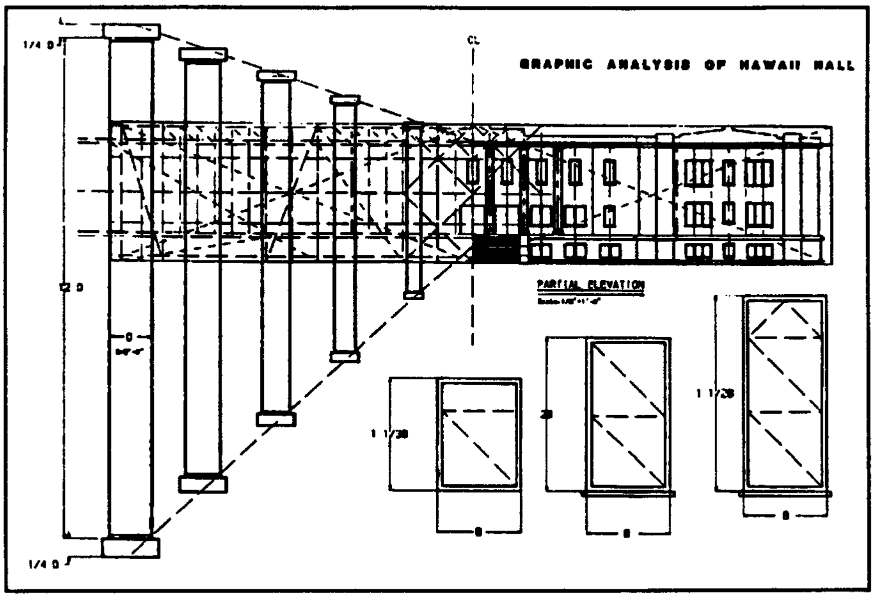

Figure 8 - By Linda Chung and Shaun Ushijima

Currently, the last assignment is to take the architectural design language and develop a new syntax for its use in application to a new elevation. All of the buildings on the selected list are essentially horizontal so the base for this final elevation is vertical. The point of this exercise is to create a design language from an existing vocabulary by focusing on syntactical issues to create form rather than the forms themselves. Figure 10 shows a "high rise" syntactical interpretation for the forms of a 2-story Korean pagoda.

A major inconsistency in the last two assignments is the lack of organized critique. The sixth exercise is essentially a preparation for the next exercise, and the last assignment is due on the last day of classes. Most students seem to be more interested in getting done with the work and back to studio than in getting feedback at this point in the semester. The course should end more with a bang than the whimper that it does, and this segment definitely needs more work.

READING AND WRITING ASSIGNMENTS

The goal of the assigned reading is to provide students with a current survey of CAD development, application, and design theory as it relates to computer applications. Many selections have been taken from published proceedings such as those from ACADIA '86 and ACADIA '87. Students have seemed to especially appreciate Doris Shaw's observations on the differences between architects and engineers[3] and Harry Wagtner's reflections on creativity[4]. For the Fall Semester 1988, readings from *Computability of*

Design have been included. A particularly useful basic reference which is being required for both computer courses in the School of Architecture is *CADD Made Easy*[5]. It is an introduction to computers written for architects which raises issues related to both design theory and professional practice. The book is current, and much of the material will probably remain current for some time to come. Another book whose text and graphics reinforce design language theories is *Architecture: Formal Approach*[6].

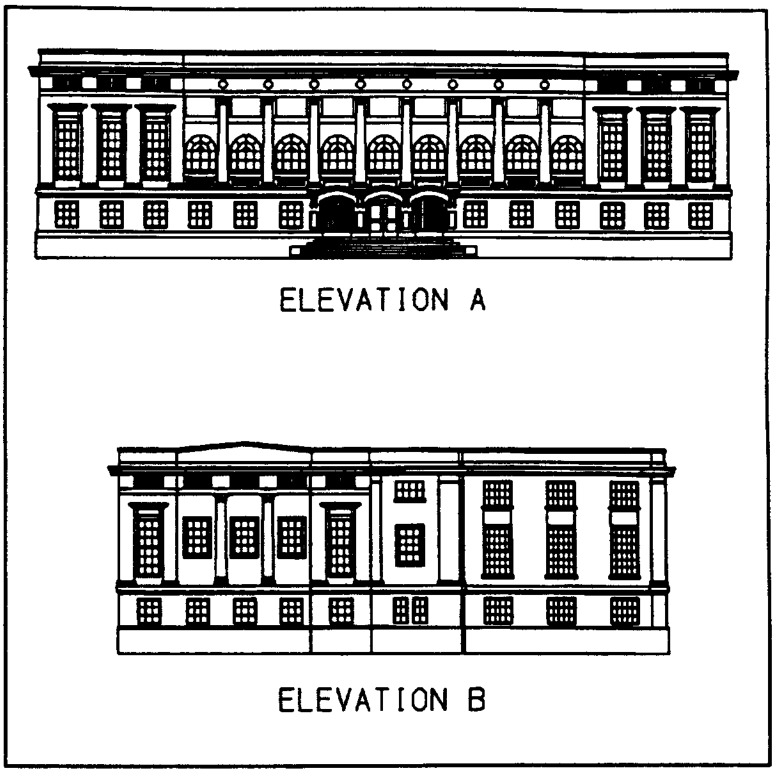

Figure 9 - By Cindy Work and Janna Mihara

Three writing assignments are given during the semester in which students are expected to respond to an excerpt from an article they are not likely to have seen. Students have about a week to write six to ten typewritten pages. For the last two semesters, I have used the concluding paragraphs from Mitchell's 1974 article in *Planning and Environment B*[7] for the first writing assignment. These assignments are an integral part of the course, more important in some ways than the computer work. It is important that architects be able to express themselves in writing, as well as public speaking, in part because the ability to think critically about various issues is enhanced through writing. This course attempts to get student to not only think and do, but also to document their experience in writing.

AN EVALUATION

What could be done differently that would produce a better result? There are the limitations of a semester schedule, but perhaps more could be accomplished within the semester. Another non-graphic exercise has been incorporated into the course for the Fall 1988 Semester which will concentrate on building mechanical skills as a foundation for the design exercises.

Students have noted the need for some useful documentation for system commands and operations. Prime's documentation is poorly laid out and confusing; less than useful for classroom instruction. However, new software to be installed soon is supposed to have a CAI module. Students may be required to proceed through the tutorial on the system according to a schedule to be set.

A critical need lies in the 3-D modeling area. Students are really excited about this facility, but its demand on processor resource makes extensive use of this module impractical. The School plans to acquire new equipment, but constraints of funds and facilities make this action problematic in the near future. The most obvious alternative would be a micro-computer based system which uses a totally three-dimensional database. In fact, the requirements are relatively limited: 3-D database, object oriented, ability to nest objects hierarchically, and basic graphics systems commands.

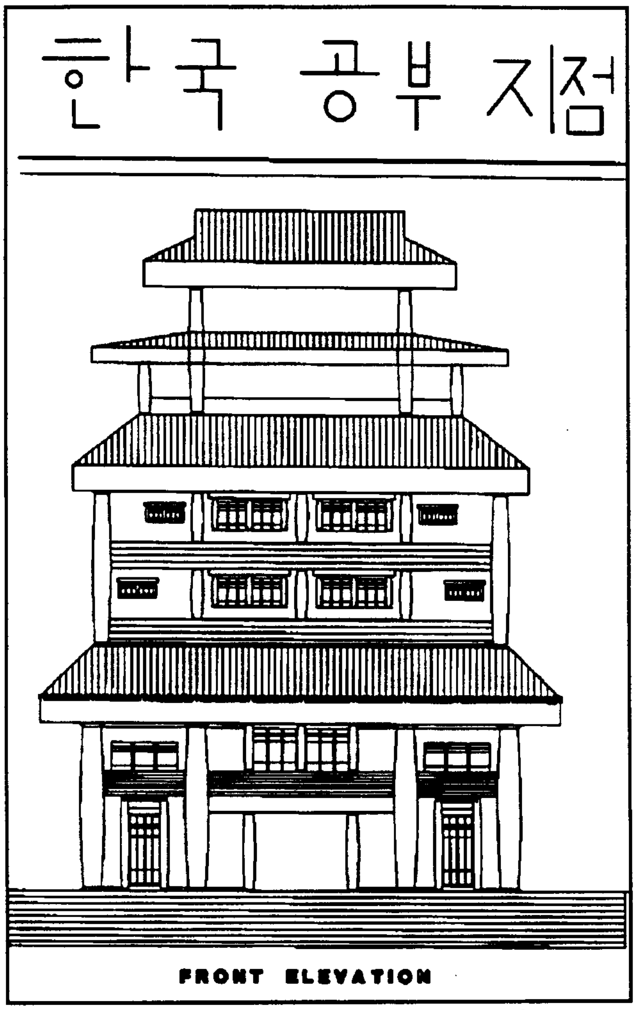

Figure 10 - By Joseph Chan, Kent Sakata, and Stacey Tangonan

However, some things are working. All undergraduates are required to take this course; not many schools are requiring such exposure at the undergraduate level. Some students are enthusiastic enough to ask to do studio projects on the system, putting up with all the deficiencies in order to put the tool to use within the context of their own agenda (see Figure 11). The positive response may be turning into a negative with more demand than the capacity can possibly serve.

A strength of the course is that it is not system dependent. The course could be taught on PC AutoCAD, Intergraph, or MacVersaCAD with few modifications. When new equipment is acquired, the course should be relatively easy to "port" to the new system. Through the course, it is emphasized that the design ideas being taught could be implemented on a variety of systems which students may encounter.

Some of the students who complete the course are using the concepts of design language theory in their studio design process, and then wrestling with the conflicts that occur with other design theories. It gives them an intellectual basis to use when they begin to formulate their own critical systems. This theoretical framework is reinforced by using it in the jury environment. This brings the computer application out of the "computer room" and into the studio environment where it has validity independent of the electronics.

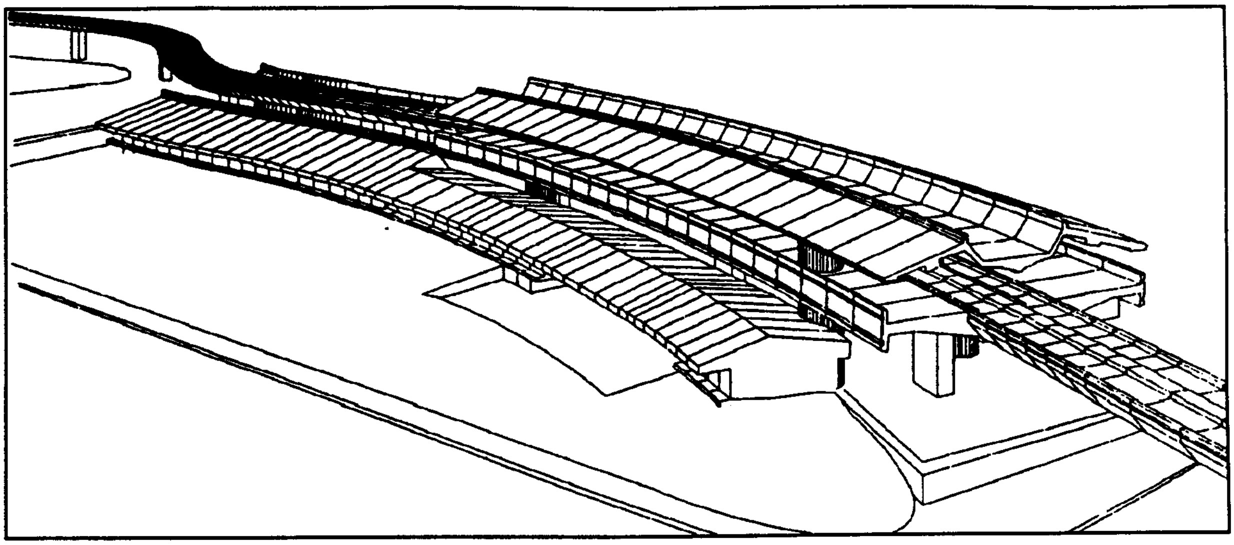
Figure 11 - By Eric Morgan

CONCLUSION

The methodological approach taken has been clearly in the "tool-using" arena. However, the context within which students learn how to use that tool should make them aware that computer graphic applications in architecture are not solely limited to technical problems and computer-aided drafting. The issues raised in the course will confront students at some point in their professional futures. It is too early to tell if these ideas are having a significant impact on our students, much less on the profession. However, if there is to be a significant impact, then students must be exposed to these ideas as a part of the basic requirements for a first professional degree.

REFERENCES

1. The National Architectural Accreditation Board (NAAB) does not require institutions provide computer courses or access to computer systems. However, "computers and information systems/software" are a specified component of the physical and information resources which must be addressed in the Architectural Program Report submitted to NAAB. *NAAB Criteria and Procedures*. National Architectural Accreditation Board. Washington, D.C. 1988.

2. Summerson, Sir John. *The Classical Language of Architecture*. MIT Press: Cambridge, Massachusetts. 1985.

3. Shaw, Doris Smith , PhD. "Case Studies in Architectural CADD Education." In *Architectural Education, Research, and Practice In the Next Decade*, pp. 157-172. Edited by James A. Turner. ACADIA, 1986.

4. Wagter, Prof. Ir. Harry. "Stimulating Creativity By Using Computers." In *Architectural Education, Research, and Practice In the Next Decade*, pp. 149-153. Edited by James A. Turner. ACADIA, 1986.

5. Radford, Antony and Garry Stevens. *CADD Made Easy*. McGraw-Hill Book Company: New York. 1987. Reviewed by Mark Clayton of California Polytechnic State University in the June, 1988, issue of *ACADIA Newsletter*.

6. Wojtowicz, Jerry and William Fawcett. *Architecture: Formal Approach*. St. Martin's Press: New York. 1986.

7. W. J. Mitchell, "The Theoretical Foundation of Computer-Aided Architectural Design." In *Environment and Planning B*, 1975, Vol 2., pp. 147-48.