Stumbling, backtracking, and leapfrogging: two decades of introductory architectural computing

Mark J. Clayton, and Guillermo Vasquez de Velasco

Our collective concept of computing and its relevance to architecture has undergone dramatic shifts in emphasis. A review of popular texts from the past reveals the biases and emphases that were current. In the seventies, architectural computing was generally seen as an elective for data processing specialists. In the early eighties, personal computers and commercial CAD systems were widely adopted. Architectural computing diverged from the "batch" world into the "interactive" world. As personal computing matured, introductory architectural computing courses turned away from a foundation in programming toward instruction in CAD software. By the late eighties, Graphic User Interfaces and windowing operating systems had appeared, leading to a profusion of architecturally relevant applications that needed to be addressed in introductory computing. The introduction of desktop 3D modeling in the early nineties led to increased emphasis upon rendering and animation. The past few years have added new emphases, particularly in the area of network communications, the World Wide Web and Virtual Design Studios. On the horizon are topics of electronic commerce and knowledge markets. This paper reviews these past and current trends and presents an outline for an introductory computing course that is relevant to the year 2000.

Keywords: *Computer-Aided Architectural Design, Computer-Aided Design, computing education, introductory courses*

Introduction

On any journey of exploration a backward look can achieve several objectives. First, it can be a cautionary action that records an escape route in case forward progress becomes untenable. Second, it can be an encouragement in that the view down the mountain shows how far one has come so that the journey up the mountain seems more achievable. Third, the view backwards may engender memories that should be recorded and preserved for future journeyers. We have journeyed far enough on the path of architectural computing education to profit from looking backward. The view backward reveals stumbles in which we have headed in the right direction but without enough caution or care. It also reveals dead ends from which it was necessary to backtrack to make progress. Finally, there have been a few dramatic leaps that foreshadow events that were in the future. This paper summarizes the record of introductory courses in architectural computing over the last twenty years.

Forward progress remains our goal. The paper concludes with a look toward the future and recommendations for the content of introductory architectural computing courses in the first years of the 21st century.

Background: computing generations

The history of computing is often placed into a framework of generations that represent convulsive changes. Although a framework of generations is usually applied to the development of hardware, it can indicate the discontinuities in development along other fronts. Table 1 lists five generations of computing from several broad standpoints.

The final generation in the table is speculative, yet the trends are clearly established. Future historians may date this generation at around 1995 or 1996 rather than 2000. Hardware will soon overwhelm our needs for processing power through some technology such as multiprocessing or further miniaturization. The form of computers is moving to something strangely amorphous. On a daily basis, most of us seamlessly use multiple computers for file serving, printer serving, mail serving or even primary processing. The computer is the network, to borrow a phrase from Sun Microsystems' advertising. Software programming is moving toward the ideal of tested, reliable components that can be plugged together and tailored to a particular need. Immersive user interfaces have already emerged from the laboratories. Software updates and even original license purchases are currently being delivered using the Internet. Soon, it may be possible to rent software components or buy processing services on the Internet.

Architectural computing has also gone through generations. Early use of computers in architecture centered upon data processing. Drafting and graphics emerged from those earlier experiments as commercial opportunities for general-purpose tools. The personal computer led to an explosion of interest as hardware and compilers became available to anyone. The Graphic User Interface of the Apple Macintosh pushed architectural computing into a new emphasis upon graphic design and image processing. The task switching and data integration capabilities of the Macintosh enabled a more fluid way of working that led to a profusion of cooperating applications. Desktop 3D, rendering and animation has provided new career opportunities for architecture students. The World Wide Web changed architectural computing again. New careers arose for our students, who are well equipped in visualizing information networks, composing compelling graphic designs, and implementing them as HTML documents. Emerging now are the concepts of "Internet-enabled CAD" in which design software communicates seamlessly with remotely located collaborators (Regli 1997). The next few years of architectural computing are likely to include another sea change toward global markets

	Year	Hardware	Form factor	Programming	User Interface	Software delivery
1	1951	Vaccum tube	Building	Hardware switch	Switch	Built-in
2	1959	Transistors	Room, mainframe	Stored program, assembly	Batch	Custom
3	1964	Integrated circuits	Minicomputer	High level languages, structured programming	Command line, interactive	Shrink-wrapped
4	1970	Very Large Scale Integration	Microcomputer	Object-oriented programming, 4th generation languages	Graphic User Interface	Internet delivery
?	2000?	Quantum? Molecular? Multiprocessing?	Distributed computer?	Components? Distributed software?	Immersive interfaces?	Distributed delivery?

Table 1 (left). Generations of computer technology.

and knowledge markets (Vasquez de Velasco and Clayton 1998).

As architectural computing educators, our key question is "what do we need to teach our students now to be ready for this future?" Looking back upon the topics that we have taught in the past can help to identify those that should be continued and those that should be abandoned to make way for new topics.

The historical record

Textbooks of the last twenty years have been examined to reveal the topics that have formally been deemed important. The list of books that were examined is provided in the references. The list of texts is certainly not comprehensive, nor is it objective. It is entirely expedient in that these books were readily available to the authors from private collections or University libraries.

The books were reviewed by perusing the tables of contents and leafing rapidly through the chapters. Many of the books were read in total, but the objective was to gain an impression of the content, biases, and emphases of each book. The following topics were distilled from all of the books as a comprehensive list of topics presented in architectural computing courses. They are listed in approximate chronological order of their emergence as topics of interest.

- **Design methods.** Design process, operations research.
- **Knowledge systems.** Expert systems, rule-based systems, shape grammars, predictive and evaluative systems.
- **Computing theory.** Discrete mathematics, logic, information theory.
- **Future.** Speculations on future systems.
- **CAAD theory.** Impacts on the profession, historical view of the profession.
- **Computer literacy.** Hardware concepts, software concepts, operating systems.
- **Business applications.** Word processing, spreadsheets, database management systems.
- **Programming.** Data structures, algorithms, control structures, structured programming, object-oriented programming, event-based programming.
- **General computer graphics.** Raster graphics, filtering, collage, vector graphics.
- **CAD management and practice.** Archiving, purchasing, field issues, training, budgeting, project management.
- **2D modeling applications.** Drafting, point line drawing, region modeling, blocks, dimensioning.
- **3D modeling applications.** Constructive solid geometry, boundary representations, rendering, lighting, materials.
- **Hypermedia.** Links, HyperCard, multimedia, presentations.
- **Animation.** Key frames, tweening, moving objects, moving viewpoints.
- **Communications.** E-mail, telecommunications technology, collaborative design.
- **World Wide Web.** HTML, browsing, authoring, plug-ins.

The summary is presented in Table 2. The textbooks are sorted by year of publication and the topics are sorted chronologically by first appearance. In addition, each book was classified as a theory-oriented text or a practice-oriented text, with the distinction that a practice-oriented text must include substantial content on the operation of specific commercial application software.

No books that were published in the decade between 1977 and 1987 were reviewed. This lack is probably due to the limitations of the library resources available to the authors, rather than a lack of publications. To address the gap, the authors reviewed several papers from the year 1983 that address architectural computing education. Foque, Hashimshony and Schijf, in outlining the origins of eCAADe, suggested that a curriculum should include

Topics	Negroponte 1975	Cross 1977	Mitchell 1977	Radford and Stevens 1987	Mitchell, Liggett and Kvan 1987	Reynolds 1987	Schmitt 1988	Crosley 1988	Mitchell and McCullough (1st edition) 1991	Jacobs 1991	Von Wodtke 1993	Mitchell and McCullough (2nd edition) 1995	Brown and Charles 1995	Sanders 1996	Woodward and Howes 1997	Bertol 1997	Kolarevic 1998
Design methods	■	■	■				■				■						
Knowledge systems	■	■	■	■		■	■			■							
Computing theory	■		■		■				■		■	■			■		
Future	■	■		■										■	■	■	
CAAD theory	■	■	■	■		■	■	■	■			■	■	■		■	■
Computer literacy			■	■	■	■		■	■			■		■	■	■	
Business applications			■			■			■		■	■	■	■	■		
Programming				■	■		■										
General computer graphics				■	■		■		■		■	■	■	■	■	■	
CAD management and practice				■		■		■	■			■	■	■	■		
2D modeling application					■	■	■	■	■			■	■				■
3D modeling application						■		■	■	■		■	■				■
Hypermedia									■		■	■				■	
Animation									■			■	■			■	■
Communications											■	■	■	■	■		
World Wide Web												■		■			
Theory	■	■	■	■		■	■		■		■	■		■	■	■	
Practice					■			■		■			■				■

Table 2 (left). Comparison of topics that are included in a sample of texts on architectural computing.

1) the fundamental concepts underlying computer methods, 2) the use of software that has been developed in the schools, and 3) programming (1983). Laing described key topics of 1) learning to use existing software that addresses small optimization problems, 2) the design and specification of software for architects, and 3) an optional topic of programming (1983). He discouraged educators from instruction in the use of commercial drafting software. While on the surface, the curriculum at MIT included a greater emphasis upon graphics, the focus was also upon programming (Purcell 1983). Education in architectural computing was presented as integral to architectural education at any level. Bridges presented a contrasting idea that architectural computing education should come only after architectural education (1983). He suggested that both research software and commercial software was inappropriate for use in education and suggested a curriculum that is essentially similar to a master's degree in software development.

Analysis of the record

The reviewed texts and the sample of papers from the early 1980's provide a reasonably complete picture of architectural computing education [A]. From this picture, one might hazard some guesses about what represents progress, even if stumbling, what directions required backtracking or abandonment of a "garden path", and what ideas leapfrogged over the contemporary computer technology to point toward something in the future.

In general, educators have explored new emphases and expanded the breadth of topics. The table illustrates these patterns graphically, as the dark cells tend downward and toward a greater quantity as one moves across the time scale. There also appears a trend toward the incorporation of more practical instruction and perhaps less theory.

One can discern some lag between the introduction of a technology and its incorporation into texts. For example, in spite of its introduction in the early 1980's, personal computer CAD software did not become an important part of textbooks until the late 1980's. Nevertheless, when one accounts for the delays inherent in publishing, our community has performed impressively in incorporating fast breaking computer technologies into our courses and texts.

The topics that have a long pedigree of continuous or recurring use are the stumbling successes of our curricula. CAAD theory and general computer graphics have steadily been addressed in texts and curricula. There also appears a continuous need to address very basic and broadly applicable topics, such as computer literacy and business applications. It may still represent a "stumble" that business applications have rarely been addressed in "how to" books for architects. CAD management and practice appears to be of growing importance as practitioners and educators gain an understanding of what it takes to adopt and manage digital tools and resources.

Some topics appear to have faded from earlier prominence. These could be considered "garden paths" that represent mistaken appraisals of what is important for architects to know about computing. Alternatively, some educators may feel that these topics should be revived. Design methods, out of which computer-aided design was born, is now rarely discussed. Computer programming briefly received much attention in the mid 1980's, perhaps as compilers and graphics toolkits for personal computers became available. However, programming has faded in emphasis and now is generally treated only as theory. Knowledge-based approaches and expert systems received attention during the heyday of those technologies, but have faded from consideration.

Most prominently missing from early texts is a thorough treatment of computer graphics from a user's viewpoint. Computer graphics, in many people's view, is synonymous with computer-aided design. In spite of occasional disdain for computer-aided drafting and an insistence that "CAD" stands for "Computer-Aided *Design*", computer-aided drafting has been the workhorse of practical architectural computing. CAD systems, even with pedestrian origins in 2D drafting, have provided a foot in the door in architecture firms. 2D CAD has then led to adoption of 3D modeling, rendering and animation. With its adoption by the entertainment industry, computer graphics have clearly been the most significant topic to emerge from architectural computing education.

In all fairness, one should also address the successes of architectural computing education. Innovations coming from our community have leapfrogged beyond the commonly accepted state of the art. The early advocates of design methods and knowledge systems may yet prove themselves prescient. The Athena Project at MIT included recognition of architecture as deeply concerned with information processes (Purcell 1983). In urging for a research discipline of information technology in construction, a recent article suggests that architects and designers are engaged wholly in information processes as distinguished from material processes (Björk 1999). The introduction of virtual design studios in the early 1990's predates the spread of the World Wide Web and suggests potentials for distributed computing and electronic commerce in design services that have yet to be fulfilled (Cheng et al. 1994).

Projections

The look at the generations of computing technology helps to identify the new topics in architectural computing education. Some of the trends for which we must prepare our students are:

Distributed computing and electronic commerce. In the near future, design services will appear using Internet-based delivery. Architects will have the option of collecting expert advice from a service on the network that may be automated. In the spirit of programming exercises in the courses in the 1970's and 1980's, our students can experiment with electronic architectural commerce by implementing simple programs.

Convergence of telecommunications and computing. Internet-based collaboration tools are currently being fielded in industry. Achieving familiarity with such tools will be an objective of future courses.

Real-time navigation through spatial simulations. Advances in computer graphics have made real-time animation commonplace. All students in the 21st century must be fluent in manipulating perspective projections and navigating through spatial simulations.

Time-scaled animation of phenomena. Real-time graphic animations provide the foundation for time-scaled animations of phenomena, such as structural deformation, construction sequences, or even energy flows.

Automated drafting and 3D architectural documentation. Drafting is rapidly becoming formalized and automated. While instruction in the conventions of drafting will still be necessary, the mechanics of producing orthographic projections are rapidly being automated.

Introductory computing should provide a broad overview of how computers can be applied in architecture. Coupled with proven and reliable course content, these trends lead to the following components of a syllabus in introductory architectural computing.

Computing basics. As qualification for a first course in architectural computing, students must show a basic computer literacy. This probably should be seen as remedial education. Entering architecture students must know the basic hardware parts of a computer and know the comparative performance capacity of an entry-level and an advanced machine. They should understand distinctions between data and software and be competent in using an operating system to manage files. Students should know how to use a word processor, a spreadsheet, a database system, e-mail software, and a Web browser.

CAAD theory and management. The course should present a brief history of computing and its application in architecture. Issues in fielding and managing computers in practice should be included.

Business computing. An introductory course should provide students with examples of how to apply standard business applications to problems typical of architectural design and management. The focus could be upon accounting, correspondence, specifications writing, and synchronous collaboration. Design methods could be incorporated into this topic of instruction.

Graphics theory. Students should be introduced to the concepts and terminology of computer graphics software, with emphasis upon Cartesian coordinate systems, affine transformations, and drawing structure. They should learn fundamentals of raster graphics and image editing.

3D modeling. The course should introduce students to 3D modeling with one or two commercial software tools. Concepts of lighting, materials and rendering should be introduced. 2D drafting should be explained as filtered projections from the 3D model.

Multimedia authoring. The use of computer-based presentation tools is becoming ubiquitous. Students should learn how to prepare graphic presentations using the computer, such as walk-through animations, presentation graphics, and desktop published documents.

Programming for integration. The record suggests ambivalence toward computer programming. However, there may still be a place for computer programming in introductory computing. Since the advent of the Web and distributed computing, scripting for integration has rapidly achieved a remarkable combination of ease of use and power. The application of programming with an information technology orientation may be an immensely useful skill to the architect of the future.

Introductory architectural computing maintains a difficult position in an architecture curriculum. There is a continuing perception that computing can be absorbed into the mainstream of instruction and thus introductory architectural computing eliminated as a separate course (Mitchell and McCullough 1995). As computing becomes pervasive throughout the curriculum, there will be increased pressures upon students to have a solid foundation that can be applied in other courses. Although introductory architectural computing is offered at the sophomore level at Texas A&M University, that is probably too late. Several freshman courses already introduce computer software for e-mail, Web browsing and even 3D modeling. Furthermore, to maintain consistent expectations of students in college level courses, there should be some responsibility placed upon secondary education to prepare students adequately. There is likely to be a need for remedial instruction to provide all students with a common level of knowledge without burdening those who already have substantial skills. A separate introductory architectural computing course, offered at the freshman level, is likely to be an important part of architectural education for the next decade.

References

B-C. Björk, Information technology in construction: domain definition and research issues, CIDAC, 1(1):3-16, 1999.

A. H. Bridges, Education and training in CAAD, in: Proc. of the International Conference on Computers in Architecture, PARC 83 (Online Conferences Limited, Middlesex, UK, 1983) 373-384.

G. Vasquez de Velasco and M. J. Clayton, Integrating introductory CAAD courses and upper level electronic design studios, in: H. Neuckermans and B. Geebelen, eds., Computers in Design Studio Teaching (ECAADe, 1998) 157-164.

N. Cheng, T. Kvan, J. Wojtowicz, D. van Bakergem, T. Causas, J. Davidson, J. Fargas, K. Hubbell, W. Mitchell, T. Hagakura, and P. Papazian, Place, time and the virtual design studio, in: Anton Harfmann and Mike Fraser, eds., Proc. of ACADIA 94, Reconnecting (The Association for Computer-Aided Design in Architecture, 1994) 115-131.

R. Foque, R. Hashimshony and R. Schijf. A framework for European CAAD Education, in: Proc. of the International Conference on Computers in Architecture, PARC 83 (Online Conferences Limited, Middlesex, UK, 1983) 339-346.

L. W. W. Laing, CAAD for undergraduate students of architecture: teach what, and when and to whom - and why? in: Proc. of the International Conference on Computers in Architecture, PARC 83, (Online Conferences Limited, Middlesex, UK, 1983), 347-356.

W. J. Mitchell and M. McCullough, Digital Design Media, 2nd edition (Van Nostrand Reinhold, New York, 1995).

P. A. Purcell, A strategy for computer education in a school of architecture, in: Proc. of the International Conference on Computers in Architecture, PARC 83 (Online Conferences Limited, Middlesex, UK, 1983), 357-371.

W. C. Regli, Internet-enabled computer-aided design, IEEE Internet Computing 1 (1997) 39-51.

Notes

[A] The following texts were used in the comparison (in chronological order by date of publication):

N. Negroponte, Soft Architecture Machines (The MIT Press, Cambridge, MA, 1975).

N. Cross, The Automated Architect (Pion Limited, London, 1977.

W. J. Mitchell, Computer-Aided Architectural Design (Mason/Charter Publishers, Inc., New York, 1977).

A. Radford and G. Stevens, CADD Made Easy (McGraw-Hill Book Company, New York, 1987).

W. J Mitchell, R. Liggett and T. Kvan, The Art of Computer Graphics Programming (Van Nostrand ReinholdCompany, Inc., New York, 1987).

R. A. Reynolds, Computing for Architects (Butterworth & Company, London, 1987).

G. Schmitt, Microcomputer Aided Design for Architects and Designers (John Wiley & Sons, New York 1988).

M. L.Crosley, The Architect's Guide to Computer Aided Design (John Wiley & Sons, New York, 1988).

W. J. Mitchell and M. McCullough, Digital Design Media (Van Nostrand Reinhold, New York, 1991).

S. Jacobs, The CAD Design Studio (McGraw-Hill, Inc., New York, 1991).

M. von Wodtke, Mind over Media (McGraw-Hill, Inc., New York, 1993).

W. J. Mitchell and M. McCullough, Digital Design Media, 2nd edition (Van Nostrand Reinhold, New York, 1995).

K. M. Brown and C. B. Charles, Computers in the Professional Practice of Design (McGraw-Hill, Inc., New York, 1995).

K. Sanders, The Digital Architect (John Wiley & Sons, New York , 1996).

C. Woodward and J. Howes, Computing in Architectural Practice (E & FN Spon, London, 1997).

D. Bertol, Designing Digital Space (John Wiley & Sons, New York, 1997).

B. Kolarevic, Architectural Modeling and Rendering (John Wiley & Sons, New York, 1998).

Mark J. Clayton, and Guillermo Vasquez de Velasco
Department of Architecture
Texas A&M University
mark-clayton@tamu.edu