Computer-Aided Architectural Design Teaching: a Design Oriented Post-Grad Experience

Abstract

CAAD education methods are often system-oriented. Software choices have an important role to play in this trend. However, teaching methods must also be taken into consideration.

An increasing emphasis has been placed on design-oriented teaching approaches in the last twenty years. By this we mean teaching methods aiming the application area in which computers are to be used, rather than the system’s structures or computer underlying paradigms. The results are mostly courses within programmes adopting design-oriented teaching methods.

However, the growing interest in this approach has rarely affected CAAD teaching programmes as a whole. A glance at the syllabus of some CAAD programmes may reveal their system-oriented nature.

We describe here a post-graduate programme that has been structured under a design-oriented approach through a set of courses in which the emphasis falls on the application in the architectural design process rather than on the software paradigms or categories.

Introduction

The idea of integrating the teaching of design computing into the design studio is not new. For many years there have been several proposals and reported experiences into this issue. Nevertheless, although substantial progress has been made in the attempt of integrating digital media and traditional ones, this issue has remained an important subject of discussion until quite recently (Dvorak, 1995; Fuchs and Martinico, 1995; Kalisperis, 1996; Kenzari, 1996; Marx, 1998)

However, most of the reported experiences deal with isolated courses within larger programmes. This approach has rarely affected teaching programmes as a whole. They rarely tackle issues relating the introduction of computing and macro aspects of design methods. By these we mean those features that are commonly accepted in the design theory of today as surpassing the specifics of particular design strategies and being common to most design processes. In this respect we believe that the macro design features as described by Rittel (1972) and later validated by others such as Lawson (1980, 1994) are still valid today. They were the result of the criticism to previous attempts of systematising design. Among those it would be important to mention the non-monotonic character of the design process: every formulation of the design problem corresponds to the formulation of a solution. Design problems have no definitive formulation, that is, at any time a formulation is made, additional questions can be asked and more information requested. Any design solution is also appraised on a large number of ill-defined and conflicting criteria. As a result the design process has no terminating pointing: it could always lead to an endless sequence of feedback loops. See Rittel (1972) and Lawson (1980, 1994) for a more detailed discussion on these issues.

On the other hand, a teaching programme with well-defined courses tends by nature to induce exactly the opposite: the observation of well-defined steps and the serialisation of the design process. The fact that the skills become available only progressively through
the programme also discourages the students to handle many aspects of design at the same time. They are also discouraged to reverse decisions taken in previous stages of a design project. This is particularly applicable if those decisions took place in different modules or courses or were delivered by different teachers.

Therefore, the predominant approach, at the level of programme as a whole, remains system oriented. Courses like "Computer Graphics", "Image Processing", "3D Modelling", "Multimedia", "Web Mastering" and "Database Management" are run along side with other design specific courses such as "Architectural Design Studio", "Design Representation", "Construction Systems", without mixing with each other. We do not deny that a great progress has been achieved in some of the experiences mentioned above. Many have been the reported implementations of "Digital Design Studios" or "Integrated Design Studios" where the teaching of computers were brought into parts of the design process. What we argue here is that most of this experiments are limited to courses or few of them within programmes. They very rarely tackled programmes as a whole (see Bridges, 1999, as an example).

Context

This paper describes an attempt to apply an application-based approach to a whole programme. We obviously do not claim an innovative premise. As it was said, the idea has been out there for many years. What we claim is to have made a small contribution in terms of implementing the idea at the level of a programme.

A post-graduate programme in computer-aided architectural design was structured with this purpose from 1998 at the Faculty of Architecture an Urban Planning of the University of Brasilia. The programme was organised around courses related with the types of application of computer systems in the architectural design process.

Difficulties

The first recognised difficulty in this attempt was that not all relevant subjects could fit into a particular design project due to the nature of their contents. For example, subjects like "Knowledge-based Management Systems" tend to be useful as a background support for several design projects rather than to fit in a particular one. The learning of this subject will very often involve the implementation of a particular knowledge domain. However, in real life those systems will be useful in a particular design project if specific domains, implemented by specialists, are already available to designers at the outset of the design task. Another example could be the "Introduction to Software Engineering an Programming" which although relevant in the training of a future specialist in CAAD is very difficult to fit in a specific design project.

The second recognised difficulty was that a large proportion of prospective students still lacked major computer knowledge and skills, even at post-graduate level. We have undertaken previous attempts to introduce computers straight into the design studio at both course and programme levels. These attempts were not very successful because for many students with no computer background it turned out to be impossible to acquired basic skills while at the same time to develop highly demanding design tasks. A compromising formula needed to be found.

A third set of difficulties is related to the already mentioned tendency of a programme with well-defined courses to induce the serialisation of the design process. Also in this set is the possibility of providing skills only progressively which induces the students to focus on only one design aspect at a time and to be discouraged to reverse decisions taken in previous stages of a design project.

An implementation of a design-oriented programme

The solution was then the development of a programme composed of three main parts: the first part was dedicated to the introduction of basic skills and theoretical background such as "introduction to computing", "introduction to computer-aided design" and "design theory and computers". The second was dedicated to the development of a design project. This was the part of the programme where full integration of computer techniques and the design studio takes place. The third part was devoted to advanced topics such as "Knowledge-based Management Systems", "Introduction to Software Engineering an Programming" and dissertation writing.

This paper concentrates in the second part of the programme mentioned above. In this part, the teaching of concepts and several computer paradigms take place within the context of conceiving and developing an architectural design project. Through a set of
courses named as “common theme unit”, a design project is developed on a specific theme which acts as a learning catalyst, directing the discussion to the most relevant architectural issues. The result is a set of courses such as “Conceptual Design”, “Visualisation Studies”, “Environmental Studies”, “Structural Feasibility”, “Design Communication to the Client” and “Design Communication to the Construction Site” in which the emphasis falls on the application in architectural design.

**Brief content’s description**

We briefly describe below the contents of the three main parts of the programme as well as the courses particularly to establish the context of the second part:

**I. Theoretical introduction and development of skills**

1. Introduction to Computing: Introduction to computing concepts and technology. Introduction to hardware and software, interfaces, computer networks and the Internet.


**II. Common Theme Unit (Design Project)**

5. Conceptual Design: 3D Modelling as a design medium. Design Project. Exploration of design alternatives regarding form, sun lighting, artificial lighting and materials through 3D modelling and rendering.


7. Visualisation Studies 1: Introduction to computer animation techniques as a means of design study and presentation. Design Project.

8. Structural Analysis: Introduction to computer-aided structural pre-dimensioning as a means of facilitating the interaction of architects with structural engineers. Design Project.


12. Project Presentation: Presentation and exhibition of design projects.

**III. Advanced Topics**


**The structure of the second part of the programme**

The “Common Theme Unit” structure is described in the figure below. The second row represents the courses of the unit in dark boxes as they are delivered. This is the backbone of the programme. We insisted in labelling them with names of design issues rather than individual names.
with names of software categories. We believe that this helps to keep the focus on design issues rather than on computer paradigms. The first row describes the major computer-related contents delivered at each course, but they are supplemental to the design issues rather than what drives each of the courses.

Particular attention was given in trying to cope with the problems related to the third set of difficulties identified earlier: the induction to serialisation of the design process and the possibility of providing skills only progressively. The order of the design issues discussed are only relatively important, because once introduced many of them will become recurrent during later stages of the design process. We have tried to induce feed back loops in the process by mixing teachers with different design preoccupations and by making all the subjects already introduced assessable at each stage and at the end of the unit. The third row through the eighth one indicates a decreasing weight in the participation of each recurrent issue in the assessment of the current issue. However, they all equally contribute to the assessment of the Project Presentation at the end of the design process.

Conclusions

We are at the present moment concluding our second experiment with the above implementation. We believe we have made a small contribution by implementing an idea that has been around for quite a while in a particular way, that is, at the level of a programme. We think this strategy is promising and that the next stage would be to implement a more systematic process for assessing its results probably making use of value-added techniques.