DESIGN PROCESS SIMULATION IN DESIGN COMPUTING TEACHING: THE EVOLUTION AND ASSESSMENT OF THE EFFICACY OF A HYBRID PROBLEM-BASED APPROACH

Neander F. Silva
neander@unb.br
Ecilamar M. Lima
ecilamar@unb.br
Universidade de Brasília
Faculdade de Arquitetura e Urbanismo
Laboratório de Estudos Computacionais em Projeto, LECOMP

Abstract

Teaching Design Computing in the context of a design task is an idea derived from the problem based learning, PBL, movement. However, a PBL curriculum has rarely been implemented in architectural education, let alone in Design Computing education. The development and implementation of a feasible PBL Design Computing curriculum is described in this paper. The idea of teaching design computing through design-based approaches has been increasingly adopted over the past 15 years. However, most of the resulting experiments have been limited to specific courses within larger programmes. This approach has rarely affected a dedicated design computing curriculum. We describe here a post-graduate programme that has been structured around a specific design project through a set of courses in which the emphasis falls on the needs of the design process rather than on software categories. We also describe the preliminary results of the assessment carried out to verify the efficiency of the adopted approach and developed model.

Resumo

O ensino de computação em projeto no contexto de tarefas de projetação é uma idéia derivada do movimento de aprendizado baseado em problemas, ABP. Contudo, currículos baseados em ABP foram raramente implementados em ensino de arquitetura, muito menos em ensino de computação em arquitetura. O desenvolvimento e implementação de um currículo baseado em ABP e factível é descrito neste artigo. A idéia de se ensinar computação em arquitetura através de abordagens baseadas em projetação tem sido crescentemente adotada nos últimos 15 anos. Contudo, a maioria dos experimentos resultantes disto foram limitadas a disciplinas específicas dentro de programas maiores. Esta abordagem tem raramente afetado um currículo especializado em projeto de arquitetura assistido por computador. Neste artigo, descrevemos um programa de pós-graduação que foi estruturado em torno de um projeto de arquitetura específico através de um conjunto de disciplinas em que a ênfase recai nas necessidades do processo de projetação e não em categorias de softwares. Descrevemos também, neste artigo, os resultados preliminares da avaliação realizada para verificar a eficiência da abordagem adotada e do modelo desenvolvido.

Introduction

The assumption that teaching Design Computing by simulating true-to-life design tasks will bring better learning is not new. It is neither specific to Design Computing teaching nor to architectural education. This assumption actually stems from the broader movement of problem-based learning, PBL, which started more 30 years ago in medical schools of North America and spread into many professional fields (Boud and Feletti, 1997).

The task of defining PBL precisely is not an easy one. There are many possible strands and the boundaries with other approaches are often blurred. For the sake of the argument presented in this paper we take it as defined by Boud and Feletti (1997:2): “an approach to education” where teaching takes place by “presenting the problem as a simulation of professional practice or a real life situation”. However, we do not subscribe entirely to all its claimed features, but we adopt the one which we believe is the most relevant, not only for Design Computing teaching, but also for architectural education. Among the most contentious issues of PBL is the idea that teaching should always start and develop through presenting problems to the students (Margetson, 1997:39) rather than concepts. Although we seek a PBL oriented approach at our school, we do not agree with this dogmatic requirement and we insist in the need for hybridisation with traditional teaching methods such as lectures, tutorials and essays.

Architectural education, in contrast with many other professional fields, contains the most relevant PBL feature: the simulation of professional practice through the design studio. This feature is derived from the origins of architectural education in tutelage and apprenticeship to a practitioner. However, PBL in architectural courses is usually confined to the studio itself. It does not affect or interact with the teaching of other subjects in the curriculum (Maitland, 1997). The adoption of a PBL-like approach in specific courses within traditional programmes is not a difficult issue. The challenge becomes evident only when the goal is to simulate true-to-life design tasks across the course subjects of the whole curriculum.

We argue here that similar problems are present in a Design Computing post-grad curriculum. Several have been the PBL experi ences into Design Computing teaching (Goldman and Zdepski, 1987; Kalisperis, 1996; Marx, 1998; Johnson, 2000; Rügemer and Russel, 2000; Wyeld, 2001). However, most of them deal with specific teaching modules, are applied within the boundaries of the design studio itself or try to integrate computing into an existing curriculum (Juroszek, 1999) rather than causing actual changes to its structure.

The nature of the design process

If the objective is to simulate a true-to-life design task across an entire programme, or at least most of it, then an understanding of design theory, particularly the design process, becomes extremely relevant to the curriculum structuring. The macro features of design processes, as described by Rittel (1972, 1980) are generally accepted today. His arguments have been later adopted by others, such as Lawson (1980), Cross (1984), Goel (1995), and Cross et al (1996). For the sake of our argument in this paper, we would like to stress here 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 feed back loops.

The nature of a traditional curriculum

If a design process is inherently non-monotonic, on the other hand a traditional curriculum with well-defined courses or modules induces the fragmentation and serialisation of the teaching process. It also prevents the integration of teaching of different subjects into one design process. It encourages the students to focus on what is being currently taught discouraging the handling of multiple-criteria design reasoning.

The story of a PBL Design Computing Programme

This paper describes an ongoing post-grad teaching experience in which we have sought to overcome these contradictions. The assumption is that a PBL Design Computing curriculum can be implemented by introducing enforced recurrence. This assumption stems from the main stream design theory (Rittel, 1972; Lawson, 1980; Cross et al, 1996, and many others). However, as mentioned earlier, we never ruled out the possibility of a hybrid approach.

This project was developed so far in three phases, each of them resulting in a new curriculum model developed during the search for a feasible PBL Design Computing programme. We describe these phases on the following sections.

Phase I: a linear model

The first phase was developed and implemented in academic section of 1998/1999. Its model was linear and it was in open contradiction with the stated goals. However, it provided the basis for identifying problems and proposing new hypothesis.

The idea of starting with problems on the programme's outset was challenged from the start: it was evident the need for other types of delivering knowledge and skills at the beginning of the course, particularly the need for lectures.

Phase II: the introduction of enforced recurrence

The second phase was implemented at the earlier part of the academic section of 2000/2001. As a consequence of what was observed in the Phase I, the curriculum was divided into three parts: the first one was dedicated to introducing basic knowledge and skills. It started with plain lectures and progressively turned into PBL-oriented approach, but resorting to other teaching methods whenever needed. The second part, called Common Theme Unit, was dedicated to a major PBL experiment where students developed a full length building design and across different teaching modules. Hybridisation was again used whenever needed. The third part of the curriculum was dedicated to advanced topics that by their own nature were difficult to integrate in a specific design task, such as for example Intelligent Systems in Architectural Design. However, even here a PBL-oriented preoccupation was kept as a goal within individual modules.

This paper is dedicated to the second part of the curriculum, that is the Common Theme Unit. Its model was based on the idea of using assessment to enforce recurrence and multiple-criteria design reasoning.

Each module was supposed to introduce its specific content and then assess the resulting product against its own criteria and those of previous modules. This structure was an improvement over the previous model, but mixing assessment criteria of a module with those of previous ones was hard to implement within each module itself. The main reasons were, firstly, the resistance of some teachers to the idea of sharing his or her criteria with the ones of previous modules. The second reason was the lack of an overall understanding of the proposed approach by part of the academic staff .

Phase III: enforced recurrence and a integrated digital studio

The third phase of this project was developed at the later part of the academic section of 2000/2001. A third curriculum model, shown in figure 1, was developed with specific modules to promote recurrence and multiple-criteria design reasoning.

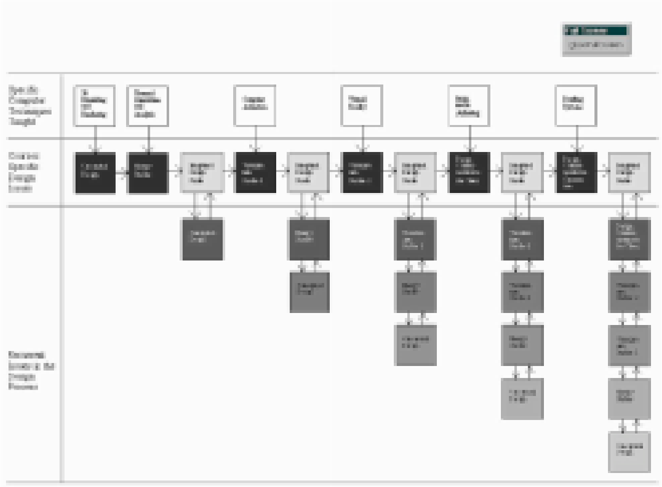

Fig 1 - The third curriculum model.

The second row in Figure 1 represents the actual teaching modules that are based on pre-selected design issues. The first row simply highlights the main computer techniques introduced in each module, but the leading element at each module is the pre-selected design issue.

This structure produced better results than the second model. At the later modules was possible to observe that students and teachers were being able to carry out some sort of multiple-criteria design reasoning, by seeking to revise previous decisions as a result of assessing their projects against more than one single criterion. However, the experience was only partially successful due to its introduction in the middle of an ongoing academic section.

Phase IV: the full implementation of the third model

Phase IV is the present step in our project. In spite of the shortcomings of the previous phases, the strategy, that is, the third model, was fully implemented during the 2001/2002 academic section, when was possible to verify its feasibility and to begin to assess some of its benefits. Figure 2 bellow shows examples of a student's work involving the application of enforced recurrence.

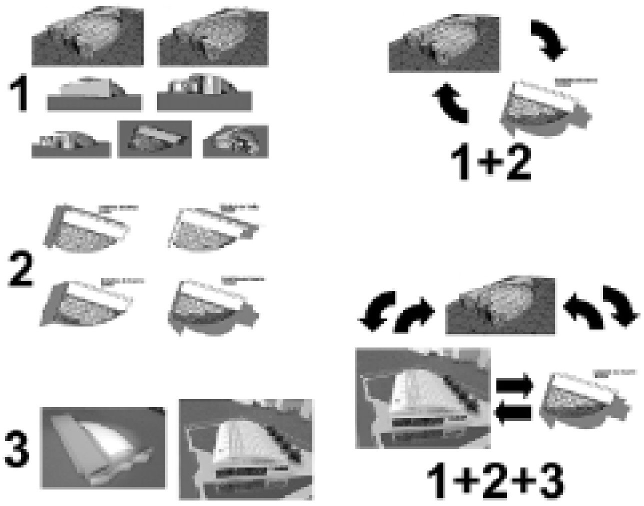

Fig 2 -. A student's work as examples of enforced recurrence

Assessment

In Phase IV of this project we used a method for assessing the performance of the adopted approach and the developed model that was based on value added (Fitz-Gibbon, 1996; Cave et all, 1997).

The students acquisition of knowledge was monitored through three exams with the same criteria. In the charts bellow, Figures 3 through 6, we show some of the preliminary results of this assessment.The data is consolidated acording to overall progress, general computing progress, computer graphics progress and design methods with CAAD progress.

Two students performances are compared here. Student 1 is around 28 years old, and has graduated as an architect 5 years ago. Student 2 is around 40 years old, and has graduated as an architect 20 years ago.

Considering their overall progress, in Figure 3, Student 1 represents the top performance, and Student 2 the bottom performance.

We are still analysing the data we have collected. However, as can be seen from the graphs above, it seem evident that the approach and model are promising, with students showing a substantial increase in their knowledge and skills.

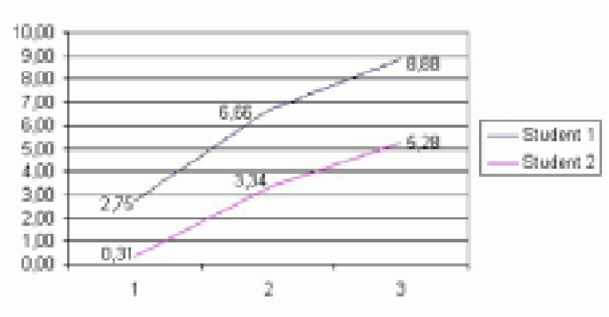

Fig 3 - Overall progress

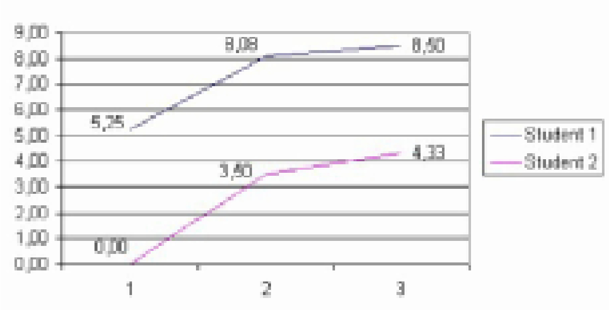

Fig 4 - General computing progress

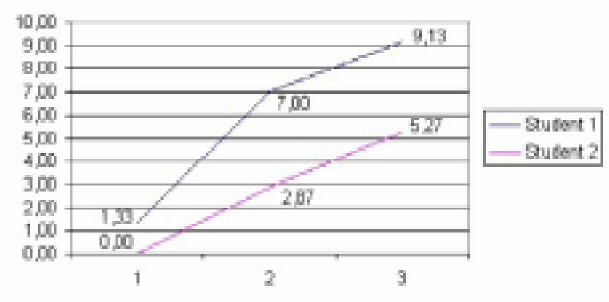

Fig 5 - Computer graphics progress

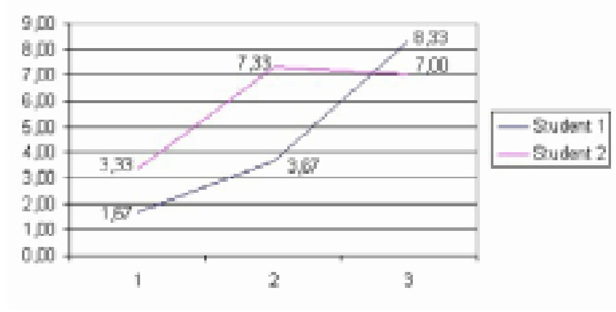

Fig 6 - . Design Methods with CAD progress

Conclusions

We believe that we have made a contribution to knowledge by developing a hybrid novel model, yet PBL-oriented, post-grad Design Computing curriculum. We believe that this model may be also useful to architectural education if PBL-oriented curricula are to be developed in this field. We acknowledge that the approach and strategy need to be more systematically assessed. We think the continuous use of value-added assessment techniques will help us to achieve a clearer understanding of the model's implementation problems and strengths.

References

- Boud, D. and Feletti, G. (1997) The Challenge of Problem-Based Learning, Kogan Page Ltd, London, UK.
- Cave, M., Hanney, S., Henkel, M., and Kogan, M. (1997) The Use of Performance Indicators in Higher Education, Jessica Kingsley Publishers, London, pp. 124-137.
- Cross, N., Christiaans, H. and Dorst, K. (1996) Analysing Design Activity, Wiley, Chichester, UK.
- Fitz-Gibbon, C. T. (1996) Monitoring Education, Cassell, London, pp. 119-132.
- Goel, V. (1995) Sketches of Thought, MIT Press, Cambridge, Massachusetts.
- Goldman, Glenn and Zdepski, Stephen (1987) Form, Color & Movement, Integrating Computers into the Architectural Curriculum in ACADIA Conference Proceedings, Raleigh, North Carolina, USA, pp. 39-50.
- Johnson, B. R. (2000) Sustaining Studio Culture: How well do Internet Tools Meet the Needs of Virtual Design Studios?, in Donath, D. (editor) Promise and Reality, eCAADe proceedings, Bauhaus-Universität, Weimar.
- Juroszek, S. (1999) Access, Instruction, Application: Towards a Universal Lab, in Brown, A., Knight, M. and Berridge, P. (editors) Architectural Computing: from Turing to 2000, University of Liverpool, UK.
- Kalisperis, L. (1996) CAD in Education: Penn State, in ACADIA Quarterly, volume 15, number 3 (summer): 22-25.
- Lawson, B. (1980) How Designers Think. Oxford, UK: Architectural Press.
- Maitland, B. (1997) Problem-based Learning for Architecture and Construction Management, in Boud, D. and Feletti, G. (1997) The Challenge of Problem-Based Learning, Kogan Page Ltd, London, UK.
- Margetson, Don (1997) Why is Problem-based Learning a Challenge?, in Boud, D. and Feletti, G. (1997) The Challenge of Problem-Based Learning, Kogan Page Ltd, London, UK.
- Marx, John: 1998, A Proposal for Alternative Methods for Teaching Digital Design, in Wyk, S. Van and Seebohm, T. (Editors), Digital Design Studios: Do Computers Make a Difference?: 59-73, ACADIA'98 Proceedings, Québec City, Canada.
- Rittel, H.W.J. (1972) Son of Rittelthink, in Design Methods Group Occasional Paper number 1.
- Rügemer, J. and Russel, P. (2000) Promise and Reality: The impact of the Virtual Design Studio on the design and learning process in architectural education, in Donath, D. (editor) Promise and Reality, eCAADe proceedings, Bauhaus-Universität, Weimar.
- Wyeld, T., Woodbury, R. and Shannon, S. (2001) Leitmotif Cases for Design Learning, in Gero, J., Chase, S. and Rosenman, M. (Editors), CAADRIA 2001, University of Sydney, Faculty of Architecture, Australia.