An Innovative Design and Studio-based CS Degree

Michael Docherty, Peter Sutton, Margot Brereton, Simon Kaplan
Information Environments Program, School of CSEE
The University of Queensland, Ipswich QLD 4305 Australia
***first.lastname*@infenv.csee.uq.edu.au**

Abstract

The University of Queensland has recently established a new design-focused, studio-based computer science degree. The *Bachelor of Information Environments* degree augments the core courses from the University's standard CS degree with a stream of design courses and integrative studio-based projects undertaken every semester. The studio projects integrate and reinforce learning by requiring students to apply the knowledge and skills gained in other courses to open-ended real-world design projects. The studio model is based on the architectural studio and involves teamwork, collaborative learning, interactive problem solving, presentations and peer review. This paper describes the degree program, its curriculum and rationale, and reports on experiences in the first year of delivery.

1 Introduction

Almost immediately after the first computer programs were written, we began to wrestle with the problem of how best to teach what became computer science (CS). Building even simple systems was quickly recognized as very difficult, and finding the correct approach or perspective could potentially provide a platform from which to build systems more easily. Briefly, the discipline has gone through two major stages: *CS as Science* and *CS as Engineering*. The names imply the pedagogical heritages from which the discipline has drawn. The scientific approach attempted to teach CS in a highly theoretical way, and drew on the notion of the scientific method and essential rationality of approach - as the foundations for an approach to teaching CS that found expression in areas such as formal methods and program synthesis. The engineering approach tries to make CS an applied discipline. The notable feature of this approach is an attempt to make the rigor of the scientific approach real-world applicable, primarily by attempting to apply the scientific approach on a larger scale, and using a more project-based approach to show that larger scalability is indeed possible.

While the engineering approach is clearly a successful evolution of the scientific, significant problems with teaching CS - and indeed understanding the essence of CS as a discipline - remain. The essence of the problem is that both these approaches assume that CS, like traditional science or engineering is a tame [9] discipline. Tameness is characterized by the separability of problem and solution spaces, and the existence of stopping rules. In other words, we can clearly specify a problem independent of any solution, and then solve that problem, with a clear guide that indicates when it's time to stop (in other words, an optimal solution has been found).

Unfortunately, CS is just not a tame discipline. Instead, it is wicked [9]. With wicked problems, in opposition to tame problems, problem and solution are inseparable, and there are no clear stopping rules. Thus, rather than being able to specify a problem completely and then solving it, one must proceed iteratively, with each attempted solution throwing new light onto the problem. Because there are no clear stopping rules, one stops when a satisficing [11] or 'good enough' solution is found. Wickedness is hardly a phenomenon unique to CS - it is common to all design disciplines. This made us stop and question whether an approach based on the pedagogies employed in design disciplines, such as architecture and industrial design, particularly the model of studio, critiques and iterative problem-based learning, would not form the basis of a better approach to teaching CS. This, then, is the unique feature of the new Information Environments degree - treating computer science, not as engineering or science, but as in its essence a design discipline.

The Bachelor of Information Environments degree focuses on the emergent areas of Interaction Design, Multimedia, the Web, Computer Supported Collaborative Work and Information Appliances, all within the framework of a sound background in traditional CS expertise.

The name "Information Environments" was chosen to avoid any preconceptions. Any other name, such as 'Interaction Design', would not only have limited the program but also constrained external perceptions of what the program was doing. "Information Environment" is a term used to describe the concept of a networked, pervasive and information rich setting in which we work and play.

Stream	Semester 1	Semester 2	Semester 3	Semester 4	Semester 5	Semester 6
Computer Science	Introduction to Software Engineering (Java)	Introduction to Information Systems Discrete Mathematics	Algorithms and Data Structures	Computer Networks	Information Science	Operating Systems
Design	Visual Thinking	Interactive Media	Human Computer Interaction	Interaction Design	Information Visualisation (elective)	
Information Environments	Introduction to Information Environments		Project Lifecycle		Advanced Information Environments	
Studio	Studio 1	Studio 2	Studio 3	Studio 4	Studio 5	Studio 6 (major project)

Figure 1. Course streams in the Bachelor of Information Environments degree. Elective slots are available in semesters 4, 5 and 6.

This goes beyond the traditional concepts of information systems and user-interfaces; the emphasis is very much on the way in which people interact with technology.

The remainder of this paper is organised as follows. Section 2 looks at the curriculum of the Bachelor of Information Environments degree. The studio-based approach to teaching and learning is described in section 3. Section 4 outlines the lessons learned to date in the offering of the degree and section 5 concludes the paper.

2 Bachelor of Information Environments Curriculum

The Bachelor of Information Environments (BInfEnv) degree is a three year degree (four years with the optional honours year) which consists of streams of courses in four main areas: Computer Science, Design, Information Environments and the integrative Studio stream. Figure 1 shows how the streams of courses are spread across the six semesters of the degree.

The program is much more tightly integrated than a traditional CS degree with a large number of compulsory courses. This allows the studio courses to assume particular knowledge and prerequisites on the part of the student, better supporting the consolidation and integration aspects of the studio stream.

The CS, Design and Information Environments streams of courses are described in the sections below. The Studio stream of courses is described in section 3.

2.1 Computer Science Stream

The CS stream of courses consists of the core programming, systems and information-systems courses from the mainstream CS degree. The courses have been chosen so as to provide students with the essential implementation skills and fundamental knowledge necessary to understand and be able to perform the design and development of information environments.

2.2 Design Stream

The design stream of courses is a unique feature of the BInfEnv. This stream includes the courses Visual Thinking, Interactive Media, Interaction Design, and Human Computer Interaction (HCI). The design stream introduces and develops a range of skills that designers need that one does not typically find in technical degree programs. These skills fall into four areas: generative thinking skills, observation skills, production skills and integration skills.

Generative skills

Students in technical fields at university often display strong skills in verbal reasoning, symbolic manipulation, logic, and reductionist problem solving. However designers need other skills. For example, it is important to generate hundreds of design ideas in order to populate the design space with candidate solutions, before beginning to judge and cull them. Non-designers are often uncomfortable working in this expansive thinking mode and like to judge ideas as soon as they are proposed, rather than playing with them, mutating them and exploring possibilities. The Visual Thinking course develops students' ability to generate ideas, to expand the space of their thinking, to identify their implicit assumptions and cultural blocks. It aims in part to develop an attitudinal shift in students. The Visual Thinking[1] course develops generative skills in the context of developing the essential design skills of seeing, sketching and imagining [7].

Observation skills

Designers need to be good observers and listeners. We introduce skills in observation through observing humans (using techniques in ethnography [2]), through learning to draw and through improvisational drama. Designers who are insensitive to the problems and needs of other humans are unlikely to design well. Problem sensitivity is essential to and inseparable from creativity. Through using

[1] The Visual Thinking course is based on a course of the same name taught in Mechanical Engineering at Stanford University. The course has undergone a variety of mutations in order to tailor it for Information Environments students.

ethnographic techniques to observe people, students learn to see what other people do. User-centred design is introduced in the Design stream and heavily reinforced through the Studio stream. Ethnography is also introduced from another perspective in the Information Environments stream.

It is important to note here that the degree program does not expect any ability in drawing at entry and it does not particularly emphasise artistic sketching. We focus on teaching students to represent what they see and to sketch their design ideas from their imagination. Sketching is both an observation skill and a design production skill.

Production skills
Students need to be able to prototype their designs in a variety of media in order to get the ideas out of their head and into the world where they can then see them, test them and get feedback on them. Different media are suitable for testing out ideas at different stages of refinement. Because information environments are not only screen based, but can comprise anything that you can embed a sensor or chip into, we develop skills in sketching and physical prototyping as well as skills in use of multimedia software.

Students also undertake a more traditional HCI course. This introduces a number of different conceptual models to help students think through human-computer interaction issues. It also introduces requirements engineering.

Integration Skills
Design involves exploring a problem or need in many different contexts, developing understandings and resolving relationships between often conflicting factors in order to bring about a satisfactory and unified solution. Students learn to propose, develop and resolve design problems through a variety of projects. This skill is developed heavily in the integrative Studio stream (see below).

2.3 Information Environments Stream

The Information Environments stream of courses covers the history of information environments, technical basics, case studies, professional topics and social issues.

The courses emphasise the importance of the user in the design process and cover the concepts of user-centred design and ethnography. Students perform ethnographic studies of several physical environments (including workplaces) over the course of the three courses and use the gathered information in the development of software representations of the environments.

Examination of case studies is particularly important in these courses. Examples of Information Environments studied include MUDs/MOOs (Multi-user domain/MUD-object oriented) and air-traffic control environments. Students study how information is represented, what the different types of users are, how the users interact with the information environments, and other features of the environments. Other computer-supported cooperative work (CSCW) environments are also studied.

Professional topics covered include briefs, lifecycles, processes, ethics and project management.

3 Studio-based Approach to Learning

The studio stream of courses is the defining feature of the Information Environments degree. During every semester of the degree, students undertake a project-based studio course in which open-ended design problems are tackled.

3.1 The Role of the Studio

The studio-based approach to teaching and learning is modelled on the architectural studio, which encourages a community of learners to interact to solve problems. It is an immersive approach to learning where open problems are visited iteratively. Studio-based teaching has long been the norm in disciplines where the nature of practice is the development of abstract artefacts that are used by others. Studios, and the attendant teaching mode of mentor and coach, can be found wherever art, interior design, architecture, graphic design, etc, are taught. The Studio courses in this degree in each semester are designed to provide opportunities for the practical application of the knowledge and skills derived from the other courses in the degree.

The structure and content of the Studio course is based on the concept that design must be taught as a hands-on project-based course. Within the general structure and aims, staff devise suitable projects for each new semester which provide the learning vehicle for the pursuit of those aims, and which require the development of the skills which are seen within the program structure as appropriate to that level of the program. The overall planning of the course stream is co-ordinated by the Studio Teaching Group on behalf of the full staff of the Information Environments Program.

A project is a block of teaching/learning. It might be made up of a seminar series, one or more projects, a set of lectures, or any form of learning activity. A project will generally last 4 or 8 weeks. There are two projects in a semester. The projects explore: issues, context, theory, practical skills, design skills, personal communication skills, industry, technology, research and literature. Projects may also include students from other programs.

There are various ‘interim’ presentations during the semester, followed by feedback sessions in smaller groups. Projects end with a public presentation, attended by all students, all staff and researchers, and visitors affiliated with the particular Project. This presentation will generally happen during the examination period and will be followed by individual feedback sessions.

Students are expected to develop skills in the resolution of design ideas, through various media available to them. They are also expected to demonstrate a breadth of knowledge regarding existing exemplars in Information Environments and present a rationale for their approach to the design of Information Environments.

Many of the software tools that are needed by the students require intensive instruction before they are able to be applied. One or two workshop weeks are scheduled each semester, where the focus is on skills acquisition in tools for graphic design and image manipulation, web editing, animation, 3D construction, video, audio, etc. Workshops are also used for the acquisition of physical skills such as prototyping in Foamcore.

3.2 The Pedagogical Approach

The pedagogy underlying the Studio approach has its theoretical origins in social constructivism and is influenced by the work of Lebow [6], and of Jonassen et al [4][5]. It is an approach that places the learner at the core of the teaching and learning experience, and, like Sims [11], emphasises the importance of learner activity and interaction as viable mechanisms to support learning. It believes that "what the student does is actually more important in determining what is learned than what the teacher does" [10]. In a break from the transmissive mode of delivery, characteristic of many lecturing situations and increasingly evident in the design of many content-driven online courses [1], the Studio approach creates a learning environment in which students are actively engaged in the learning process. It also creates a more fluid and responsive learning environment.

The project-based approach of the Studio provides an environment for learners to:

- plan and develop their own learning projects
- apply the skills and knowledge learned in concurrent courses to the specific real life scenarios provided by the projects
- work collaboratively with other learners to develop knowledge and understanding
- work with mentors and coaches
- work as mentors and coaches

The projects provide realistic and relevant contexts which encourage ownership and voice in the learning process. They also provide opportunity for authentic assessment by assessing not a number of facts or concepts that are memorised and reproduced under examination conditions, but the learner's ability to use and apply the knowledge acquired in the learning process in the types of settings and situations where it is ultimately destined to be used. Transfer and application are important outcomes of learning in the Studio approach.

3.3 The Role of Technology in the Pedagogy

Technology is used in the Studio, not to deliver content, but to move the classroom focus from the teacher to the learner, and to create a more active and engaging climate for the learners. Oliver [8] argues that such a shift will encourage learners to develop their understandings and their capacity to think and act critically. Students are encouraged to use hardware and software to create new information environments and the WWW is used to support communicative, collaborative and cooperative activities among students and teachers.

The Studio aims to foster creativity, reflection, articulation and reasoning, all of which are important lifelong learning skills and valuable graduate attributes and the purposeful use of technology and of group work is central to these aims.

3.4 The Role of Space in the Pedagogy

Supporting this pedagogical approach is the program's commitment to the use of physical space as a resource to solve problems. Trevitt and Chalmers [13] investigate the use of learning space by students and teachers and the impact of the physical settings on teaching practice and learning activities, and their investigations include a focus on the Information Environments program. The Studio courses are held in purpose-built spaces designed to encourage collaboration and team approaches to problem solving. While there is naturally a strong emphasis on access to technology in this degree, the design of the physical space varies significantly from the rigid line-up of traditional computer labs. The Studio spaces are more fluid, allowing the formation and re-formation of different teams and different spaces according to the problems driving the student interactions.

4 Experience To Date

The Information Environments program commenced teaching at The University of Queensland at the beginning of 1999 with an intake of 30 students. The program is undergoing controlled growth with a 2000 intake of approximately 50 students and an expected 2001 intake of 70-80 students.

Communicating to the students why they need design skills was an initial difficulty in the freshman Visual Thinking course (part of the Design stream of courses). Many students initially saw the course as one of learning to sketch and to build models – akin to something that they did in high school – and a distraction from learning the technical skills that they need to program. In part, the problem arose in the first year because of our difficulty in communicating to our first year of prospective students that this was not a pure CS degree. Many students were not expecting and were resistant to learning to design. Our second year intake is quite different in make-up from our first intake. In a recent survey 70% of students said that they chose the degree either (a) because of its design component or (b) because it was not a "straight [CS] degree." Having also changed the curriculum to link the sketching skills more obviously to design project work, 70% of students expressed in a mid-semester survey that they were enjoying learning to sketch and see.

Students have reacted favourably to the studio-based approach and this comment is indicative:

> "The thing that I like about the projects is the ability to pull together a lot of different technical

> skills. If you're just working on Java like they do in other courses, then you only look at Java in that course. When you have a [course] like Studio which actually draws on all [courses], you can use the Java aspect with some of your design skills and you get to team together a whole lot of things and not only work on one course skill. This is where you get to really pull everything together and say 'oh I have to think about that', where if you're doing just one [course], you don't have to think about how it reacts or interacts with anything else, and I find that really excellent."

This approach has required teaching staff to take on new roles and teach in ways that are much less teacher-directed and much more focussed on the centrality of the learners and their learning. It has also resulted in greater collaboration across the degree structure and greater willingness on the part of teaching staff to be more responsive to capturing teaching and learning opportunities as they arise as opposed to rigidly delivering a pre-planned program of work.

The multidisciplinary nature of the teaching staff and the use of team-teaching in some courses (especially the studio courses) has proven to be an important factor in the success of program. The teaching staff have backgrounds in computer science, architecture, mechanical engineering, industrial design, electrical engineering, cognitive science and psychology. The varied perspectives that these staff bring to the program, and in particular to the studio projects, have resulted in a better educational experience for the students.

5 Conclusion

This paper has described an innovative degree program being offered by The University of Queensland. The Bachelor of Information Environments is a design-based CS degree which augments the core content of a Computer Science degree with a stream of design courses and integrative studio-based projects undertaken every semester. Graduates from the Information Environments degree will have a solid computer science grounding and an extensive range of design skills.

Evaluation studies have indicated that students in the Information Environments degree program overwhelming endorse the Studio-based approach to teaching and learning which is at the heart of the pedagogical design of this degree. Students welcome the opportunities the Studio projects provide them to contextualise the skills they've learnt in other courses and to apply them to real-life design problems. They also value the group work associated with the projects, and comment on the importance of reflection and feedback to skill development and the growth of their conceptual knowledge.

The Information Environments program, although young, has demonstrated that the architecture-studio model of teaching and learning is applicable, and indeed successful, in the field of computer science. The studio-based learning combined with an emphasis on design has resulted in an innovative CS program. It is expected that courses and teaching techniques from this program will find their way into the more traditional CS programs over the years ahead.

References

[1] Brown, A. (1997) Designing for learning: What are the essential features of an effective online course? *Australian Journal of Educational Technology* Vol. 13, No. 2, summer 1997, pp 115-126..

[2] Fetterman, David, M. (1998) *Ethnography : Step by Step* (Applied Social Research Methods Series, Vol 17), 2nd edition.

[3] Hooper Woolsey, K., S Kim, G Curtis (1996) *VizAbility Handbook and CDRom*, PWS Publishing.

[4] Jonassen, D., Mayes, T., & McAleese, R. (1993) A manifesto for a constructivist approach to uses of technology in higher education. In T. Duffy, J. Lowyck and D. Jonassen (Eds.), *Designing Environments for Constructivist Learning* Berlin Heidelberg, Springer-Verlag.

[5] Jonassen, D., & Reeves, T. (1996) Learning with technology: Using computers as cognitive tools. In D. Jonassen (Ed.), *Handbook of Research on Educational Communications and Technology* New York, Macmillan.

[6] Lebow, D. (1993) Constructivist values for instructional systems design: Five principles toward a new mindset. *Educational Technology Research and Development*, 41 (3), 4-16.

[7] McKim, Robert H (1980). *Experiences in Visual Thinking* by Robert H. 2nd edition, Brooks/Cole Pub Co.

[8] Oliver, R. (1998) Teaching and Learning on the World Wide Web: A Guide for Teachers Western Australia, Edith Cowan University.

[9] Rittel, H.W.J. and Webber, M.M. (1969), "Dilemmas in a general theory of planning," *Policy Sciences 4*, Elsevier Science, pp 155-173.

[10] Shuell, T. (1986) Cognitive conceptions of learning. *Review of Educational Research*, 56, 411-36.

[11] Simon, H.A. (1960) The new science of management decision, Harper and Row.

[12] Sims, R. (1999) Interactivity and narrative: Strategies for effective learning, in B. Collis and R. Oliver (Eds.) *Proceedings of EdMedia 1999.* Charlottesville, VA, Association for the Advancement of Computing in Education.

[13] Trevitt, C. & Chalmers, D. (2000) Learning, architecture, planning and design: An empirical investigation of learning spaces at The University of Queensland. Conference paper to be published in the proceedings of the ASET /HERDSA Conference, *Flexible Learning for a Flexible Society* USQ, Toowoomba, Queensland