CREATING AUTHENTIC TECHNOLOGY ENVIRONMENTS FOR IMMERSIVE LEARNING

Alan Boykiw
Information Environments Program
The University of Queensland, Australia
boykiw@csee.uq.edu.au

Allison Brown
Teaching and Educational Development Institute
The University of Queensland, Australia
allison.brown@mailbox.uq.edu.au

Abstract

This paper argues that traditional teaching methods in computer science do little to prepare students for the demands of the IT industry, where they are required to problem solve to meet the rapid changes in technological innovation, and to work in creative and responsive teams. It describes an approach taken in a new computer science degree at The University of Queensland, which is firmly embedded in learning-centred pedagogy. The educational design of the degree emphasises active, interactive and collaborative learning on the part of students. In this degree, students work as a community of learners, interacting to solve design problems. Courses provide a team-based environment for technical development and implementation, rather than the traditional inflexible, sequential and compartmentalised IT development methods found in most computer science degrees. This paper describes uses of technology that differ from current computer science education practice. Typically, technology application is thought of in terms of resources used to deliver effective teaching. In this paper we describe a course which immerses students in the evaluation, design and construction of technological constructs to achieve certain outcomes.

Keywords

computer science, technology for learning, collaborative learning, studio approaches

Introduction

The Information Environments degree at UQI represents a major departure from traditional didactic computer science teaching methods and embraces learning-centred pedagogy as a means of achieving improved learning outcomes and transferable skills. The Information Environments program focuses on the human side of computing, particularly on how people in different locations communicate and share information on networked systems and devices. The degree is a design-focussed, studio-based program, which emphasises the crucial importance of cognitive skill development in the areas of problem solving and critical thinking for computer science students. An end-user focus and teamwork emphasis are also central themes of the program.

The Information Environments program recognises that it is no longer sufficient to pursue inflexible, sequential and compartmentalised IT development methods or to focus on delivering content in relative isolation of the contexts in which it is to be applied. Brown, Collins & Duguid,

(1989) argue that this separation of content (what is to be learned) from context (how it is learned and used) will not result in deep learning. Students can ‘acquire’, and even manipulate, algorithms, routines and de-contextualized definitions, but this does not mean that they will be able to apply them to new contexts. Lave (1996) also argues that learning is a process that takes place in a participation framework rather than in an individual’s mind. Software engineering skills that have been taught in isolation result in little transfer to new contexts, and, an added complication is that, given the rapid developments in technology, the content and skills presented to students in this way are likely to have changed almost before graduates have reached their first jobs.

Brandsford, Sherwood, Hasselbring, Kinzere and Williams (1990) found that many software engineering students face difficulties when it comes to problem solving – they often have no idea where to begin, despite their familiarity with the syntax of programming languages. They can memorise facts and procedures, but have difficulty in explaining observed phenomena, or solving real-world problems or analysing problems and thinking critically. Many of these students may pass examinations, but have trouble generalising their learning from one situation to another, leading to a skills gap every time the job, content or technology changes. Future employers expect computer science graduates to be able to meet the rapid changes of technological innovation, and to work in creative and responsive teams. Traditional teaching methods do little to prepare students for these demands.

Studio-based Teaching and Learning

The majority of IT degrees typically include a single undergraduate subject concerned with user-focussed development and human-computer interaction issues. In contrast, the Information Environments program is educating IT students with a human factors and design-first pedagogy, achieved through a ‘Studio-based approach’ to teaching and learning, The ‘Studio’ enables a community of learners to interact to solve design problems. The ‘Studio approach’ offers students an opportunity to solve real design problems in ways that mirror the work of professionals in the world of information technology: through team work, collaborative learning and the application of related knowledge to new contexts (Docherty & Brown, 2000). Knowledge and skills are acquired in context rather than as separate segments ‘to be learned’. This approach closely relates to social constructivist theories of learning (Jonassen, Davidson, Collins, Campbell & Haag, 1995) which argue that learning is necessarily a social dialogical process in which communities of learners socially negotiate the meaning of the phenomena. Meaning is constructed through collaboration and conversation by the learners, rather than through the passive receipt of ‘uncontested knowledge’ delivered in a one-way flow of information by the lecturers. The emphasis on ‘Studio’ as a physical and virtual place provides a context for work on physical and virtual outcomes and deliverables. Studio courses integrate learning from design and IT courses in the program.

Studio IV occurred over a period of thirteen weeks and was the fourth Studio undertaken by second year students in the Information Environments degree. The emphasis in Studio IV is on the production of working prototypes based on research and ideas. It was designed to explore information environments from the perspective of invisible or ambient computing technology. This is in contrast to the current genre of PC workstation applications that is the dominant focus of computer science education initiatives. The ambience domain was chosen to encourage students to examine new ways of communicating and interacting with information through unique technology solutions. The abstract nature of invisible computing prompted the need to provide a tangible and physical focus in order to achieve the learning outcomes mentioned above. A virtual café was chosen to be the information environment of interest for the duration of the course. Human experience was emphasised within this context. Off-the-shelf hardware such as sensors, microphones, lights, speakers and smart circuit boards were to be combined with software applications to design and build physical demonstrations of a café concept.

The students proceeded through various phases requiring both individual and group contributions to the overall project. These phases were modelled on typical IT industry practice based on the lecturers' industry experience. This course was taught by two lecturers who took roles within the scenario as development manager and client, rather than traditional lecturing roles. This entailed managing the students as a development team, inspiring them to present solutions to the 'client' as the project progressed. The initial phases of the project were conducted in groups of four students. Their tasks included ethnographic observation and interview of café user groups, physical/virtual problem and opportunity analysis, and information flow evaluation. This research of the different types of users in the café scenario provided the students with a set of needs and problems that they could design solutions. For example, owners were consistent in identifying that changing menus and 'specials' was a problem for them and was a significant cost to them.

The next phase entailed making individual preliminary design proposals, physical and virtual design mock-ups that would satisfy the user needs identified. Hardware technology was also evaluated and the students presented ways of applying them to satisfy user needs (user centred design). The designs were evaluated in terms of suitability for creating an overall café experience that demonstrated notions of ambient and invisible computing to a lay audience. The reality of constructing the designs within the allotted time for the program was also considered which is typical of industry practice. Both students and lecturers participated in a review session to determine the most feasible alternatives.

The development of the virtual café software followed a practice called eXtreme Programming (Beck, 1999, 2000), involving user story cards, pair programming, task assignment, time estimates and test case development. This is a highly learner-centred and activity-focussed approach and it was selected because of its emphasis on rapid, nimble, iterative development with a strong user focus. Students took intimate ownership of the outcomes resulting in a deeper appreciation of the rationale and application of effective design processes and methodology.

Outcomes

The final result of the students' efforts was a physical manifestation of the software code and hardware configurations that they had developed throughout the semester. The software that was developed enabled hardware such as touch and motion sensors, microphones, lights, interactive whiteboards (SmartBoards), projectors and speakers to create a unique café experience. During the evening that the café was set up for inspection, it was very evident that student learning went far beyond the classroom learning outcomes. Students fielded questions about their designs and were engaged in discussions with IT professionals reinforcing the reality of what they had researched, designed and built. In most cases, the discussions centred on user interaction and value rather than on the technical achievements. From a technical perspective, the achievement of having the students build the prototype should not be under-stated. The software that was developed by the students through this process enabled hardware such as touch and motion sensors, microphones, lights, interactive whiteboards (SmartBoards), projectors and speakers to communicate to create the unique café experience. The idea of building things in contrast to cursory code reviews appealed to the students and was reflected in their course evaluations.

Also, the students did not need to be taught to configure technology in new ways to create the effects they had researched and designed. They had an inherent desire to see the results of their research come to fruition. Other examples of technology application were: chairs signalled when they were occupied, microphones picked up the level of sound in the café and this 'ambience' was represented as graphic images on a remote sign, live camera images of guests were displayed and could be streaked and altered with the stroke of a finger on a display board.

In our experience, the impact of the students' work would have been severely minimized without the overall context of the café and information environment focus afforded by the flexibility of the Studio delivery process. The technology infrastructure associated with the campus also allowed students to pursue their passion for the project as many times they worked remotely into the night uploading and downloading code from the central repository.

Conclusion

Students in Studio IV achieved outcomes that clearly differentiate their learning experience from students taught by more didactic methods. A course evaluation was completed by the students with results in the top quartile in all categories. The majority of the comments offered by the students were positive. An enhanced ability to work in teams was one of the major outcomes, with students reporting a sense of individual satisfaction in a team context by making a contribution to a shared vision. So committed were students to successfully developing the ambient café demonstration, that they performed critical set-up and integration tasks after all assessments for the course had been completed.

There was also evidence that students' oral presentation skills improved over the course. This was as a result of making frequent, informal presentations of design ideas or development status. These presentations had authenticity, being intimately and realistically associated with the context of the task, in contrast to traditional presentations which often require students to make isolated topic-centred, high pressure, seminar-style presentations for assessment purposes. The traditional confusion over the boundary between collaboration and collusion did not exist in this course. In the spirit of real-world development initiatives, students were rewarded for reuse and sharing, contrary to typical IT learning environments where similar code raises issues of collusion and penalties.

By creating a learning environment where students could maximise their engagement in learning through discussion, clarification of ideas, considering alternatives and monitoring their own understanding. In addition they were able to compare points of view with those of others and negotiate work practices; the students were able to forget they were being taught, but were aware that they were learning. Their learning was a product of this ambient collaborative culture rather than of explicit teaching. Anecdotally, they acknowledged the importance of the teamwork aspects of the course and they expressed a preference for working on this course to the detriment of other courses. They competed positively, building on each others' ideas rather than being critical or defensive about suggestions for improvement.

References

Beck, K. (1999). Embracing change with Extreme Programming. *Computer, 32* (10), 70-77.

Beck, K. (2000). *Extreme programming explained: Embrace change*. Reading, MA, Addison-Wesley.

Bransford, J., Sherwood, R., Hasselbring, T., Kinzere, C., & Williams, S. (1990). Anchored instruction: Why we need it and how technology can help. In D. Hix and R. Spiro (Eds.) *Cognition, education and multimedia: Exploring ideas in high technology*. (pp.115-141). Hillside, NJ: Lawrence Erlbaum Associates.

Brown, J., Collins, A. & Duguid, P., (1989). Situated cognition and the culture of learning. *Educational Researcher,* 32-42.

Docherty, M. & Brown, A. (2000). Studio-based teaching in information technology. In L. Richardson and J. Lidstone (Eds). (2001). *Flexible learning for a flexible society*. Proceedings of the ASET-HERDSA 2000 Conference. (pp. 213-219). Toowoomba, Qld, 2-5 July 2000.

Jonassen, D., Davidson, M., Collins, M., Campbell, J. & Haag, B. (1995). Constructivism and computer-mediated communication in distance education. *The Journal of Distance Education*, *9,* (2), 7-26.

Lave, J. (1996). Teaching, as learning, in practice. *Mind, Culture and Activity 3,* (3). 149-64.