Some Experiences With The “Contributing Student Approach”

John Hamer
Department of Computer Science
University of Auckland
Auckland, New Zealand
J.Hamer@cs.auckland.ac.nz

ABSTRACT

We report on our recent experiences with Collis’ “contributing student approach” in two computing courses. Departing radically from traditional lecture-based teaching, the approach involves students preparing learning resources to share with other members of the class. Contributions are peer assessed, lectures become class meetings, and the course web page is replaced by a shared “wiki” collaboration tool to which all students can contribute. Consequently, students are inescapably placed at the centre of all learning activities.

The approach may form a model for higher education courses that aspire to equip students with the skills necessary to function effectively in the knowledge era.

Categories and Subject Descriptors

K.3.1 [**COMPUTER USES IN EDUCATION**]: Collaborative learning

General Terms

Human Factors, Performance

Keywords

Pedagogy, Contributing Student Approach, Knowledge era

1. INTRODUCTION

Educationalists have long been arguing the need for a radical overhaul of teaching in higher education, one that shifts the focus away from the presentation of content and toward more active learning styles that better prepare students for the knowledge era.

Although the arguments are seem compelling, change is never easy. It is not a simple matter to come up with new course objectives, assessment methods, measures of success, expectations, responsibilities, workloads, quality controls, not to mention learning material. The impact on resources can be significant, and the implications for the degree programme is often unpredictable.

Despite all that, change is inexorable. It also takes a long time, and involves a good deal of trial and error. We report here on our own trial and errors, using the “contributing student approach” developed by Collis and Moonen [5].

The contributing student approach can be described simply: rather than lecturing prepared material to students, tell them to find out about one or two topics each, and share the results with the rest of the class. Some additional elements can be added: a collaboration tool is needed, so that work-in-progress is visible; also, regular peer assessments provide both a measure of quality control and a mechanism to expose students to the full range of course material.

The approach is soundly based in education theory. We summarise the relevant theories in section 2. Section 3 describes the approach and its rationale. Our two courses are described in section 4, and the core elements in section 5. Sections 6 and 7 discuss the benefits and issues we observed with the courses, followed by some concluding remarks.

2. EDUCATIONAL FOUNDATIONS

Constructivism is the dominant theory of learning today [3]. The theory claims that knowledge is actively constructed, not passively absorbed from textbooks and lectures. It holds that a student’s ability to understand new material depends on their existing knowledge, and hence that two students may come to construct different understandings from the same learning experience. The two conflicting views may not be equally viable, and learners must be challenged to test and justify their understandings and those of others. Shared meanings are thus arrived at through a process of social negotiation rather than individual study.

Wenger’s “community of practice” theory [11] stresses further the importance of social interaction in learning. It holds that meaning is derived from a dynamic context, and that all meaning is inherently social in nature. The negotiation of meaning involves two constituent processes: *participation* and *reification*.

Participation involves taking an active part in a meaning-making process, and includes relations with others that reflect this involvement. Participation is broader than collaboration, and includes reading, making decisions, working alone, etc.

Reification is “giving form to our experience by producing objects that congeal this experience into ‘thingness’ ”. In a Computer Science context, examples of reification include naming of programming language constructs, design

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ITiCSE’06, June 26–28, 2006, Bologna, Italy
Copyright 2006 ACM 1–59593–055–8/06/0006 ...$5.00.

patterns and variable roles, but also writing a computer program to express a particular concept. The concrete nature of reification helps in communication, but it can also lead to over-simplification and confusion. Participation is essential in resolving such problems.

3. THE CONTRIBUTING STUDENT

Collis [4] lists the following attributes necessary for functioning productively in the knowledge era:

- Continuously updating and changing skills
- Using electronic networks effectively and efficiently
- Handling the mobility of services, information, workforce
- Working in multi-disciplinary and global teams
- Deriving local value from global systems
- Acting autonomously and reflectively, in socially heterogeneous settings

These attributes constitute the ultimate learning objectives of the contributing student approach, irrespective of the nominal course content.

The key idea behind Collis' pedagogy is for learners to create learning materials and share them with others. Students can contribute to the learning resources based on their own experiences, the experiences of others, and by selecting material from the world wide web. A web-based collaboration tool is used to store work-in-progress and to share course material.

In this new pedagogy, a student is expected to adopt several new roles [4]:

- a co-creator of learning materials (study resources, quiz questions, model answers, help materials for other students, lecture materials, etc.);
- a responsible selector from a variety of real-world resources;
- someone who extends, rather than just reads, the textbook and the work of others;
- someone involved in self- and peer evaluation as an assessed part of the course;
- someone who designs and builds a product with a use outside of the course.

4. DESCRIPTIONS OF THE COURSES

The approach was used in two computing courses in 2005: a CS2-level course on data structures and algorithms; and a year four course on formal methods. Both courses had a similar enrolment size of around 70 full-time students. The courses are part of a four-year professional engineering degree, and the students in each given year attend most of their classes together as a cohort.

The courses run for twelve weeks, and the teaching is shared by two lecturers, each of whom takes responsibility for a six week block. In both cases, only one of the six week blocks was taught using the contributing student approach; the other half used traditional lectures.

The CS2 course, which has been offered largely unchanged for many years, covers elementary data structures (linked and array-based lists, hashing, binary search trees) and basic algorithms (searching and sorting), plus an introduction to state space search. Assessment is based on coursework (25%), a written test (10%) and exam (65%).

The formal methods course was introduced in 2004, and in that year it followed a traditional lecture form. The first block covers an in-depth study of the Alloy modelling language [8], and the second a comparative study of formal modelling languages. Assessment was 40% coursework, 10% test and 50% written exam.

For the CS2 course, students worked individually or in teams to prepare a learning resource (which could be a report, poster, software visualisation, lab exercise, or slide presentation) on one of the four main course topics: big-O, binary trees, hashing, and state space search.

Students on the formal methods course worked in pairs to develop a case study of an Alloy model and write a report. Each team also produced a poster highlighting the significant features of their case study.

In each course, the resources were peer assessed twice (a draft and then final version). Coursework marks were awarded for the learning resource, for the quality of their peer reviews, and (in the formal methods course) for their contribution during class meetings (see section 5.3). In both courses, the group mark was shared, and individual marks were awarded for the quality of the peer reviewing. The test and exam were in a traditional format, similar to previous years.

5. COURSE FEATURES

Several software systems and organisational structures became core components of the courses. Without them, the courses could not have been run as they were.

5.1 Aropä

Aropä [7] is a web-based system for administering peer assessment exercises. It was developed to support the routine use of peer assessment in large classes, and has since been adopted by teachers in several departments, including Business and Management, Computer Science, English, Pharmacology and Software Engineering.

With each course running at least two peer assessments exercises involving 70 or more students, it was essential to have software support for uploading submissions, allocating submissions to reviewers, on-line entry of reviews, computing grades, etc.

5.2 Wiki-wiki web

A wiki-wiki-web [1] ("wiki") is a web-based collaboration tool that was used instead of the usual course web site. A wiki allows any student to create or edit any page through a browser interface. Edits could be made anonymously, but students were encouraged to register with the system so that their contributions could be identified for the purpose of allocating coursework marks[1].

The wiki was used for a number of purposes. Each team created their own space on the wiki, and used it to record work-in-progress. They were able to upload notes, program code and diagrams, etc., and to save links to useful web sites.

[1] We used the MediaWiki [2] system for both courses.

The wiki was also used to coordinate some class activities, such as signing up for teams and distributing topics among the teams. The agenda and minutes from class meetings (see below) were also posted on the wiki.

Students also had the opportunity to create a personal wiki page, to share some personal information with the class and to express themselves individually.

5.3 Class meetings

The contributing student approach greatly reduces the need for formal lectures, as most of the activity of the course is undertaken by students in their own time and place. As this was the first time we had used the approach, the scheduled lecture times were retained. The lectures, however, became "class meetings".

Class meetings provided a regular forum for students to discuss and share material. Each meeting had a formal agenda, which was placed on the wiki where anyone could add items[2]. A minute taker was elected at the start of each meeting, with the responsibility of posting a summary on the wiki by the following day. Minutes recorded the names of the students who contributed, and this information was used at the end of the course in assessing class contribution grades[3]. The lecturer became chairperson, although even this role was taken by students on occasions[4].

The meeting discussions were largely driven by the students, with the lecturer (chairperson) talking perhaps 10% of the time.

5.4 Laboratories

Each course had a weekly two-hour session available in a computer laboratory. This provided an opportunity for students to work together on their coursework, and for the lecturer to assign additional practical exercises.

We initially hoped that the learning resources developed by students would include laboratory exercises for the whole class to subsequently share. Unfortunately, this proved to be a more difficult task than expected, with the students generating draft material that was either long and tedious or too easy. In the end, the lecturer ended up setting most of the lab work.

In order to increase engagement in the laboratories, student were asked to post their results onto the wiki. After the completion of the laboratory, they were invited to look at the results posted by other students and identify any discrepancies.

For the CS2 course, a typical lab questions might look like this:

- Which of the three basic data structures (`ArrayList`, `HashSet` and `TreeSet`) uses the least memory for a collection of size n, where $n \in \{10^1, 10^2, 10^3, 10^4, 10^5, 10^6\}$
- When adding a string to a `HashSet` of size n, what portion of the time is taken up in computing the hash code? Does the answer change for different sizes n?

[2]With a wireless network now covering most lecture theatres, items would sometimes be added surreptitiously during the meeting!

[3]MediaWiki has a "what links here" facility that identifies all the pages that reference a user's personal page.

[4]Notably, during an industrial strike by academic staff the meeting went ahead without a lecturer present.

The questions are intended to create uncertainty. No experimental design is specified, so students had to decide for themselves what to measure and how to measure it. Different results arise from different design assumptions, and many subtly faulty designs are possible. Issues arising from a lab were discussed in the next class meeting, and a "lab maintainer" nominated to assembling a best solution from the various fragments.

6. WHAT IT BUYS YOU

6.1 Communication

Since students are both producers and consumers of material, they get to exposed to the challenges of effective communication.

> *"I learned that producing such resource [notes] is not an easy task, because you have to consider which part of the information will need to be included and which don't. It is also hard to communicate through writing that will ensure that the reader will understand what is being written, not to go out of the topic being discussed, to keep the reader focus and not to get bored."*

The range of tasks involved is very broad, as related by the following student:

> *"From preparing this resource, I've learned:*
>
> - *how to abstract useful information from a large number of resources*
> - *how to summarise things I learned and understood*
> - *how to explain to others about a topic*
> - *how to research for needed resources*
> - *how to use time efficiently*
> - *how to abstract information from other students*
> - *how to compare and modify things I know with things I learned from other students' resources*
> - *how to demonstrate my knowledge*
> - *how to specify"*

Students readily saw themselves as creators of knowledge rather than passive recipients:

> *"Doing research teaches you a lot"*
>
> *"It's actually quite helpful to go and find information for ourselves."*

6.2 Different perspectives

The diversity of course material offers some advantages. The next comment hints at the intellectual development stages identified by Perry [9]. Perry claimed that students progress through a series of well-defined "positions" relating to their perception of knowledge. The earliest stage is *dualism*, in which the world is comprised of right and wrong answers. Later stages admit conflicting answers, which are initially seen as arbitrary and only later understood in terms of a broader context that requires decisions based on personal experience and reflection. The student quoted below is clearly comfortable with different perspectives on course material, although for other students this was a source of confusion.

> *"This project has taught me research techniques and I find that I have gained various perspectives on the same material as different authors have different viewpoints. I learned teamwork and also made a new friend."*

There were several occasions in which conflicting information arose in the CS2 course. The hashing topic was a particularly rich source of misinformation [6], given the number of poor hash functions in common use. Controversy was also explicitly sought in laboratory experiments, by asking students to post their (often divergent) lab results onto the wiki.

6.3 Teamwork

While the contributing student approach does not require it, most learning materials were developed by teams. This decision was made for both pedagogical and practical reasons. The ability to work effectively in a team is an important skill, requiring much practice. But furthermore, teams produce a smaller quantity of higher quality materials. With five resource types and four topics, a class of 70 students can easily cover all combinations while working in teams of between 2 and 4.

Favourable comments about teamwork were frequent.

> *"Overall in preparing a visualisation we learned a lot of coding techniques and new features that Java has to offer. We also learned big-O in depth. Also a major thing I say worked out well was the way that we worked as a team and were able to help each other out."*

> *"Through the teamwork we had plenty of conversation about data structures as well as our topic. Also through wiki-web I can see other people's idea (concept) and could learn many things which probably I could not learn from normal lecture.*
>
> *The most important thing I learned from this is heuristic. Not just learning from tutors or from books, asking myself and trying to solve what I couldn't understand is very useful to me...*
>
> *Overall, I have learned so many things that I can not learn other classes and I appreciate it."*

7. ISSUES AND PROBLEMS

A host of concerns about the course were articulately expressed by one (A-grade) student:

> *"I do not believe student directed learning was beneficial in learning and understanding the material in this course as it further promotes common falsehoods is not time efficient and does not utilise the accumulated knowledge of the lecturer*
>
> *Student directed learning although gives a good understanding in one topic falters in all others, a great deal of time is required on the students' behalf to recreate only a sub-par knowledge riddled with inaccuracies that could've been cleared up by the lecturers knowledge in the subject. Overall the costs of students directed learning are too great and this course would be much more informative in a conventional setting."*

These objections clearly reflect an *Acquisition Model* of learning [10], with its focus on the acquisition of pre-specified knowledge. The importance of course content is downplayed in the *Participation Model* underlying the contributing student approach, to the extent that "inaccuracies" and the secondary role played by the lecturer are seen as serving a positive purpose.

The student's comments indicate that more time might be spent in discussing the educational philosophy behind the course. Nevertheless, the concerns cannot be lightly dismissed. Both Acquisition and Participation are needed, and the challenge is to strike the right balance.

7.1 Fairness

Unavoidably, students receive an uneven exposure to the course material. The topics for which they prepare resources are much better known than other topics. This phenomena raised concerns about fairness, and required some care to be taken in the design of the test and exam to give equal weight to each topic.

Using the wiki as a collaboration tool meant that draft resources for all topics were always available to the whole class. However, while the students were good at updating the wiki with their own work, they tended not to spend time reading material from other groups.

The peer assessment exercises proved to be the main occasion for students to study material outside their own topic. However, in our inexperience, reviews were allocated randomly, and most students ended up missing out reviewing one or more topics. Running two peer assessments—draft and final—helped mitigate any imbalance. For future years, we plan to spread the reviewing more evenly, to ensure all students review all topics.

8. CONCLUSIONS

Collis' contribution-based pedagogy offers students a very different experience from traditional higher education. They are forced to actively contribute to an emerging community of practice, a community in which all members depend critically on the performance of each other. The approach is flexible enough to allow students room for individual expression. Their existing knowledge and skills are valued, and there are many opportunities to contribute in different ways.

It has been interesting to observe the reactions from students to an unfamiliar and potentially threatening learning environment. The loudest objections have tended to come from A-grade students, while many weaker students thrived.

Overall, the feedback we have received from students suggests that while they may not all like the approach at first, it is effective at developing a range of desirable skills.

9. REFERENCES

[1] Wiki. `http://en.wikipedia.org/wiki/Wiki`, accessed 17 January 2006.

[2] Media wiki. `http://www.mediawiki.org`, accessed 23 March 2006.

[3] M. Ben-Ari. Constructivism in computer science education. In *Proceedings of SIGCSE'98*, pages 257–261, Atlanta, GA, 1998.

[4] B. Collis. The contributing student: A blend of pedagogy and technology. In *EDUCAUSE Australasia*, Auckland, New Zealand, Apr. 2005.

[5] B. Collis and J. Moonen. *Flexible learning in a digital world: experiences and expectations*. Kogan Page, London, 2001.

[6] J. Hamer. Hashing revisited. In *ITiCSE2002 Seventh Conference on Innovation and Technology in Computer Science Education*, pages 80–83, Aarhus, Denmark, June 2002.

[7] J. Hamer. Aropä user manual. `http://www.cs.auckland.ac.nz/~j-hamer/pas-manual.pdf`, 2005.

[8] D. Jackson. *Software Abstractions: Logic, Language and Analysis*. MIT Press, 2006.

[9] W. Perry. *Forms of Intellectual and Ethical Development in the College Years*. Holt, Rinehart, Winston, New York, 1970.

[10] A. Sfard. On two metaphors for learning and the dangers of choosing just one. *Educational Researcher*, 27(2):4–13, 1998.

[11] E. Wenger. *Communities of Practice: Learning, meaning, and identity*. Cambridge University Press, 1998.