Peer Assessment for Action Learning of Data Structures and Algorithms

Philip Machanick
School of ITEE
University of Queensland, St Lucia
Qld 4072, Australia
philip@itee.uq.edu.au

Abstract

This paper describes an experience with use of peer assessment in tutorials as a tool to promote deep learning from early stages of a course on Data Structures and Algorithms. The goal was to improve the utility of tutorials in encouraging more efficient learning habits. Since assessment forms a key part of the actual curriculum, tutorial exercises were for credit, but the emphasis was on formative assessment. The novelty in this approach is that peer assessment has not been extensively studied in Computer Science Education for content of the kind covered in this course. Evaluation is limited by the fact that other details of the course were changed. Two surveys were conducted, one soon after the first assignment, the other soon after the second assignment. Of various aspects of the course surveyed, the tutorial quizzes were the least popular, but improved in popularity between the two surveys. The overall effect based on general observation of the class appeared to be positive. Results were closer to a normal distribution than for the previous 2 years. Performance in the first assignment, which required understanding of how the theory is applied in a practical situation, suggested that deep learning had taken place.

1 Introduction

> *The irony does not escape him: that the one who comes to teach learns the keenest of lessons, while those who come to learn learn nothing.*
> — JM Coetzee, *Disgrace*[1]

There is a discontinuity between techniques in undergraduate teaching and knowledge formation in research. In research, peer review is common, with the assumption that all participants are equals – if reviewers are more equal than than authors. Peer review is not an uncommon approach in the workplace, though perhaps it is more common that review is by an immediate superior. An unstated assumption is that the employee will "graduate" to the level of their superior through a process of learning on the job and occasional review.

Given that both the ordinary workplace and academia see a role for peer review or at least review by the community within which one works, there seems to be a case for a similar process in learning in formal education.

This paper reports on an experiment in introducing "peer review" in the form of peer assessment into teaching of a classical area of Computer Science, data structures and algorithms. Peer review fits the *action learning*

[1]Random House, London, 1999. p 5.

paradigm (Bunning 2001) well, as it introduces a strong aspect of reflection. Through understanding the assessment process, students should build a clearer notion of their educational goals, and be able to plan better for future assessment. Further, the process of learning should be closer to practices in the real world where design reviews, for example, are common.

While this general notion is reasonably well accepted in education, there is relatively little literature on peer assessment in Computer Science, and it is generally focused on a narrow range of areas – software projects (Ruehr & Orr 2002) and design of Internet services (Brookes & Indulska 1996) are the two most common areas. What little work there has been on peer assessment in algorithms (Hübscher-Younger & Narayanan 2003) is in a broader context, and is not specifically focused on peer assessment as a tool.

The aim of this paper therefore is to present an investigation into the value of peer assessment specifically in data structures and algorithms, but even more specifically as an aid for making tutorials more effective. Peer assessment is potentially useful for tutorials for several reasons. Tutorials without assessment are seldom taken seriously because students can be expected to focus their energies on assessable activities (Biggs 1999). Since tutorials are the most regular interaction with the class where assessment could take place, using them for formative assessment (Brown 1999) is useful. Tutorials are a natural place to do peer assessment because of the formative aspects inherent in evaluation of the work of others.

The approach which was adopted in this research was to introduce peer assessment into tutorials, with a specific goal of introducing formative assessment. Given that the course was run on a different basis with different lecturers as compared with immediate predecessors, comparison with past versions of the course is difficult. However, evaluation based on general levels of student comprehension of concepts known to be difficult, and surveys of student attitudes to the approach provide some measurement of outcomes. In addition, general observation of the behaviour of the class is some indication of the success of the intervention, if lacking in rigour. Finally, evidence of deep learning (Biggs 1999, Ramsden 1988) could be found by the use of assignment questions or examination questions which could only be answered if the students had formed their own model of the key concepts in the course.

Most specifically, the alignment of intended learning outcomes with assessment (Biggs 1999) was measured through performance in the first of two assignments, in which students were presented with a problem from a section of the course they hadn't seen yet, and had to *understand* the theory to be able to solve the problem.

The remainder of this paper is structured as follows. Section 2 contains a brief review of background and related work. In Section 3, more detail of the approach to the problem is supplied, with results in Section 4. The paper ends with a concluding section containing reflections on the approach taken and the results.

2 Background and Related Work

2.1 Introduction

Peer assessment is not a new concept. However, its use in Computer Science education is limited. This section briefly surveys some general ideas behind the paper's focus on peer assessment, while placing peer assessment in the context of action learning.

A key aspect of action learning is reflection (Bunning 2001). Since reflection is inherently looking back at yourself, it appears that reflection requires looking inwardly. However, if we think of learning as a social process, reflecting on yourself requires looking at yourself in the context of what others are doing. Peer assessment captures the essence of this mixture of looking back at oneself yet looking at others. Even when evaluating a piece of work which someone else has done, students are still likely to be thinking about what they did. But seeing someone else's work adds a social dimension to learning (Berger 1966).

This section examines these ideas further, with a view to justifying the research approach. However, before delving into the specific educational model, it is useful to give some space to the general nature of education, to have some indication of what teaching and learning are trying to achieve to provide a basis for evaluating an intervention.

Subsection 2.2 provides a starting point by defining what educations is – at least in adequate terms to relate to the intent of the intervention described in this paper. Subsection 2.3 summarizes some key ideas on how knowledge is socially constructed. In 2.4, these ideas are related to the idea of peer assessment. Aside from general work on peer assessment, Computer Science application is examined. Finally, all these ideas are put together in 2.5, which summarizes the key issues as they relate to this paper.

2.2 What is Education?

The nature of education and its societal role has changed over time. Historically, there has been a distinction between "training" – the development of job-specific or vocational skills – and "education" – the development of more abstract skills (Brookshear 1985, von Glasersfeld 1995, Moore 1998, Denning 1999, Sanders & Mueller 2000).

The development of this distinction and the relative roles of these two kinds of learning has varied with changes in society. In ancient Greece, for example, the well-off "free" citizen would expect to have leisure time to dispute with philosophers, attend dramas and think about the nature of the universe – a precursor of the modern notion of a liberal arts education: learning for its own sake. This notion of what we could call *aristocratic* learning persisted into medieval times, though with important differences. As the fraction of aristocratic members of society declined with the collapse of the Roman Empire, learning became by and large a property of the clergy, to the extent that in medieval England, to be classified as *clerici* meant that you were learned in Latin and the classics, not necessarily that you were a priest or a monk. Although there was a separate word for the literate (*litteratus*), it was more or less synonymous with *clericus*. The *illitterati* and *laici* correspondingly were the same group. In that era, the church represented the only route of upward mobility for the non-aristocrat, so there was considerable economic value in learning Latin and the classics. Further, since *clerici* were much less susceptible to being sentenced to be hanged for a felony, the ability to stand up before a court of law and recite some convincing Latin had more than a trivial social utility – so by 1300, the meaning of *litteratus* or *clericus* had shifted from a highly learned person to anyone with reasonable reading skills (with a gradual shift from Latin to the vernacular) (Clanchy 1993).

By a roundabout way, the ancient Greek tradition of education for its own sake began to acquire a utility beyond merely impressing peers at the dinner table. Showing some evidence of learning could open up what was then a lucrative career in one of the few organizations which had money, and could even save your neck in a crisis.

In recent times, the tension between "useful" and "abstract" education has grown, because of the success of higher education over the last century. An egalitarian society requires that everyone have access to a social benefit, so higher education is increasingly open to all. Yet, at the same time, the filtering aspect of higher education – the certification of individuals as being capable of something above the ordinary – requires that there be some limit to how many can be certified, before the social utility of higher education as a meal ticket breaks down. Inevitably, stress is placed on the system: grade inflation avoids the unfriendly notion that egalitarian access doesn't mean all should pass the filter, dumbing down of hard topics avoids collapse in student numbers because a subject is "too hard". Students enjoy an illusion of learning; academics keep their jobs (Adams 1980).

The problem with this development is that there *is* some utility in more abstract skills – even if the need for these skills may not be as great as the opening up of higher education suggests. In fact, the trend in industrialized countries is away from manual skills: poorer countries are more cost-effective in that realm. A move towards a more "vocational" style of education in wealthier economies is therefore likely to be self-defeating. Long-term trends in the US, for example, suggest that service jobs are growing faster than industrial jobs are shrinking. In the US over the last 10 years, services have grown 37%. At the same time, manufacturing has shrunk 11%, not because less has been sold (wholesale and retail trade grew 11% and 17% respectively). General economic non-agricultural activity over that period grew 18% – all as measured by change in number of employees in each sector (US 2003).

Whether the balance is right – too many students aiming for abstract skills versus hands-on skills, for example – is a debatable matter. However, attempting to maximize the number of students capable of operating at a more abstract level appears to be a useful goal: such students are more likely to add value to their society by being innovators – as evidenced by the increasingly rapid development of not only industries but social institutions in countries with strong higher education systems.

2.3 Social Construction and Action Learning

As access to education has transformed, models of teaching and learning have evolved from psychological notions of cognition through the philosophical ideas of phenomenology to social theories (Berger 1966). Through all of this, in the post-modern spirit of evading definition, it is not clear that a stronger notion of the nature and goals of education has emerged. Rather, the conflict between universal access and filtering remains, and is yet to be resolved. However, advances in models of education can be applied to resolving this dilemma.

The approach taken in the work reported on here is to assume that maximizing the number of students who attain a reasonable level of abstract skills is a useful goal, while recognizing that many students are expecting a more vocational training-style of learning experience. Reconciling these contradictory goals requires placing the learning of abstract skills in context, so they can be seen as a legitimate preparation for the real world.

Consequently, the work reported on here is based on the idea of *situated learning* – that learning has to be seen in the context of the social process of work (Lave & Wenger 1991).

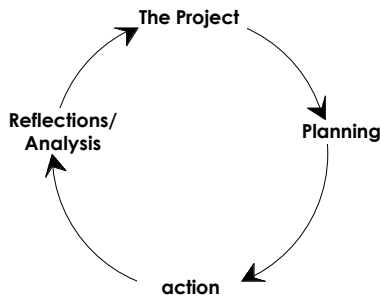Situated learning – which Lave and Wenger (1991)

Figure 1: Action Learning Cycle

prefer to call *legitimate peripheral participation* – has much in common with the ideas inherent in the social construction model of education, though it comes from a different starting point – the observation that traditional modes of learning through models like apprenticeship involve the learner in a community of practise. The community starts from a position of "superior" knowledge, and the learner is gradually brought to a position of influencing knowledge formation in the group, by active participation.

The introduction of the term "legitimate peripheral participation" (LPP) is intended to convey the notion that "situated" should not be read in a trivialized sense – learning in the workplace, for example, as meaning putting a teacher with chalkboard in a warehouse or factory. Rather, the learning is "situated" in a social context – a community of peers and "old-timers", with the interactions between the various participants shaping the learning process. In line with the ideas of social construction, there is an emphasis not only on the role of learning in shaping the learner, but in the potential for a learner to shape the community of which they form part. Newcomers start out on the periphery, but they are "legitimate" – real participants in knowledge use and formation. They gradually become integrated into the community, and join the "old-timers".

Such a notion of social construction of knowledge should fit well in an engineering environment, where it is increasingly being accepted that the theoretical underpinnings of a subject stick better if taught in conjunction with applying the principles (Director, Khosla, Rohrer & Rutenbar 1995). The effect of such improvements in teaching style is to bring the teaching practice closer to "real" engineering: students use real tools to solve real problems (Schön 1995); they see theory in the context of applying it, rather than as a burning hoop to jump through before the fun starts.

This kind of work is not well known in Computer Science education. Such action learning work as has been done has been more commonly aimed at areas like work-based learning (Bradley & Oliver 2002) and training in use of technology (Vat 2000) rather than hard Computer Science skills.

Even cognitive science-based approaches (which may seem closer to Computer Science philosophically, in that much Artificial Intelligence work has roots in cognitive science) have not gone much beyond Papert's work on relatively early learning (Papert 1993). One detailed study of the value of visualization tools in teaching Computer Science (Naps, Rling, Almstrum, Dann, Fleischer, Hundhausen, Korhonen, Malmi, McNally, Rodger & ngel Velzquez-Iturbide 2003) for example has evaluated the success of the approach in terms of matching levels of Bloom's Taxonomy – a useful starting point, but one rooted in the 1950s.

The social construction model does not in itself dictate how learning should take place. One model which can be used is *action learning*, which empasizes a cycle of starting from a desired outcome ("project"), planning an approach to solving the problem, applying the plan

("action") and reflecting on the outcome. If the problem is not yet solved, the cycle is repeated (Figure 1 (Bunning 2001)). Action learning fits Computer Science education well, because many Computer Science problems can be formulated in the style of a problem to be solved, with the possibility of forming a plan which can be tried out, and evaluated ("reflection"). The step of reflection is easy to leave out – or pay lip service to – with programming problems because there is a temptation to make random changes to the program until it appears to behave correctly. For this reason, in a data structures and algorithms course, it is useful to focus on enhancement of reflection as a learning technique.

2.4   Peer Assessment

The literature on peer assessment – while varied and presenting many viewpoints – tends to be specific to teaching a given subject.

In Computer Science education, there has been some work on using peer assessment in teaching algorithms (Hübscher-Younger & Narayanan 2003), and a tool has been devised for anonymous peer review of student work (Gehringer 2001).

Viewed in the Computer Science Education context, it is surprising that these ideas have not been taken up with more enthusiasm. Much Computer Science work is inherently social – software development, for example, is a team activity in the real world – yet team-based learning has not advanced much beyond the problems inherent in individual assessment of team projects.

Tools to support peer review are commonplace in management of academic conferences; developing such a tool for learning should not be difficult. At least one such tool, Peer Grader (PG), exists (Gehringer 2001), if reports on its usefulness are rather thin. Some reported uses include researching lecture material, researching beyond lecture material, reviewing published papers, and reviews in programming projects. The last idea is of most relevance here. In keeping with the LPP idea, design reviews are a common technique in industry. Performed as a teaching exercise in which members of a class review each other's designs, the students are exposed to a style of work which at the same time promotes learning. By seeing what the "community" looks for in assessing a design by doing the assessment themselves on another class member, students can both play the role of the learner and the "old-timer". Seeing both perspectives creates a sense of the real community in which students will work. Unfortunately, Gehringer (2001) does not analyze the potential of the approach to this level of detail, and stops at anecdotal evidence backed up by surveys of students' opinions.

A more detailed study of constructive and collaborative learning in an algorithms course (Hübscher-Younger & Narayanan 2003) presents evidence that a collaborative process (using a tool called CAROUSEL) in which students produce their own representation of data structures, and present it for peer review, is beneficial to learning. There have been many attempts in the past to produce palatable representations of data structures, involving graphics and computer animations. There have been mixed results in evaluating such attempts at making data structures clearer to the novice, probably related to factors like different learners having different learning styles, and confounding variables like teacher enthusiasm. An important factor in evaluating such tools is to ensure that learning outcomes are evaluated, not just learner enthusiasm – otherwise it is possible to arrive at positive results for use of a tool which doesn't stimulate learning (Wilson, Aiken & Katz 1996).

Peer review is an important component of the CAROUSEL tool, as is the notion that encouraging students to find their own representation. A likely reason for the mixed results of tools like algorithm animators is that

they present a pre-formed notion of the "correct" way to think of a concept, hardly compatible with the notion of a community of learners and teachers in which the learners aim to join their teachers as peers – as new "old-timers" (Lave & Wenger 1991).

Results of studies of the use of CAROUSEL show that active participation is important to learning – best results were seen with students who contributed representations, as opposed to those who only reviewed work of others (Hübscher-Younger & Narayanan 2003).

2.5   Summary

In summary, the notion of Computer Science students as members of a community of learners, who review each other, while building for themselves a role in the overall community as eventual "old-timers" has some support in existing work. Tools to support such a style of learning exist, and are not difficult to create. There is some evidence to support such a strategy. Unfortunately, the evidence remains to be tested at a detailed level: is peer assessment, for example, a good strategy on its own – or does it only work in combination with other strategies which place students in a community of learners? More specifically, how important is the notion that the community is not isolated from the real world, but rather, an extension of the real world, to which students are in effect being apprenticed? How important is it to link strategies like peer assessment to the notion that students should be aiming not merely to be knowledge absorbers, but knowledge creators – without some sudden discontinuity on graduating?

3   Methodology and Approach

The goal of the experiment reported here is primarily to make tutorials a more useful learning experience. No amount of cajoling will persuade students that they should actually arrive at a tutorial in a position where they have done absolutely everything they could do on their own without help. Instead, students commonly arrive at a tutorial hoping it will be something like a lecture – if they plead convincingly enough, they will be given the answers and allowed to go home.

The question – and related question of what to measure – addressed in this paper is:

> *How can students' expectations of a tutorial be be met as closely as possible yet changing the effect of the tutorial to supporting deeper learning? How do we measure whether that deep learning has taken place?*

An important issue from the perspective of taking this idea forward is that, while there is support for the general notion of peer assessment in the general educational literature, experience with the ideas in Computer Science Education research remains mostly anecdotal or thinly tested. There are several militating factors against relatively rigorous evaluation in this study. The course was run by a different lecturer the previous year, the programming language used in the course has changed (the change, from C++ to Java, is significant) – and there was another intervention in the same course by a second lecturer (introducing motivating examples in lectures).

Accordingly, the approach to evaluation had to rely on imprecise instruments, which limits repeatability and the strength of conclusions. The strongest measure of whether deep learning had taken place was performance on an assignment which required that the students have a good model of how the theory worked in practice. Some attempts have been made at correlating the outcome of this assignment with the tutorial intervention and other aspects of the course, though those these attempts are limited by the imprecision already noted.

Taking all the above factors into account, the approach to the intervention and evaluation are spelt out in this section.

Subsection 3.1 outlines the intention behind the intervention. Subsection 3.2 describes the intervention in more detail, followed by 3.3 which elaborates on the approach to evaluation. The section closes with overall conclusions on the approach (3.4).

3.1   Intended Effect

The intention was to promote deep learning by making tutorials more effective. Specifically, the intent was to promote the students' ability to develop their own model of meaning and their ability to apply it to a novel situation. This is in keeping with the notion that deep learning requires thought about what is really meant by what is being taught, how to relate new knowledge to old, and how to apply knowledge to a novel situation (Ramsden 1988).

It is possible to a some extent to promote deeper learning by withholding solutions, and challenging the students to produce their own. To do so however is an uphill struggle, with some members of the class insisting to the end that they are entitled to solutions and that the absence of solutions inhibits their learning.

Do they know something experienced lecturers do not know?

Perhaps – but perhaps not. If students are used to having solutions, a major shift in strategy is bound to make them uncomfortable. Since comfort factor can be a significant indicator of success in a course (Wilson & Shrock 2001), upsetting students is not a good start to having a change accepted. On the other hand, giving out solutions gives the impression that students need only wait until the "right" answer is given to them – there is no perceived value in working things out for themselves. There is no promotion of the type of deep learning (Biggs 1999) which should pull students away from what I call "binge learning" – only really working on deadlines, and forgetting much of what they learnt afterwards.

To achieve the goal, therefore of making tutorials more effective, it seems that, perversely, it is important to appear to be giving the students exactly what they want – plenty of examples, tutors willing to give out solutions, good feedback on what they are doing – but to introduce something into the process which encourages more active participation from them.

Given that assessment drives de facto student activity, ("actual" curriculum), small quizzes with peer assessment were introduced as a way of forcing the students to engage with the material and reflect on it.

3.2   Intervention

The specific approach to be used was to make tutorials relatively short sessions of working through material, followed by a quiz on the same content. Tutors had answers, and were encouraged to make them available much more readily than normal. The purpose of the tutorial from the students' perspective was intended to be preparation for the small quiz, followed by peer assessment.

Each quiz (except the first two which were weighted 0.5%) counted as 1% of the final grade. The quizzes, while not trivial, were designed to be easy enough that a student who was paying attention and who had *understood* the essential concepts should be able to achieve close to 100%. However, the quizzes were not straight factual recall: they emphasized applying knowledge, with some degree of novelty.

Tutorials were designed to provide a range of questions from very easy to a bit more difficult than this class should cope with (except the top students). The quizzes were pitched closer to the simpler than the harder level. In this way, if students could do the easier questions on

their own, and arrive at the tutorial primed to find out how to handle the harder questions, they should have been in a good position to do well on the quiz.

If they did not do well on the quiz, they still had the option of a second bite by trying again to understand what they should have done while doing the marking.

The overall effect, if it worked as intended, should have been that the students worked slightly more continuously, and developed a deeper understanding earlier in the course.

3.3 Evaluation

Evaluation was difficult because there are potential confounding variables, as described already.

Direct evaluation of the effect of the intervention on tutorials could be measured by monitoring tutorial attendance. Quiz results and tutorial attendance could be correlated with other assessments, to attempt to determine the effectiveness of the intervention on learning.

Another intervention in lectures also occurred, which makes it difficult to evaluate the effect of this intervention in isolation. For assignments, tutorials are likely to have a stronger impact than lectures on improvement. Again, there is no baseline as a basis for comparison.

Student surveys are a useful instrument to measure student perception of the value of the intervention. However, measures of learning outcomes are also important, as student perception is not always accurate (Wilson et al. 1996). Accordingly, the first of two assignments was designed specifically so that students needed to have a deeper insight into the theory and how it applied than is usual for a course at this level. The second assignment and examination were more typical of assessments of previous versions of the course. The notion was that the first assignment should demonstrate whether deep learning had taken place if its results correlated more strongly with the tutorial quizzes than the quizzes correlated with other assessments.

The remainder of this subsection describes the measures used: student perception, followed by student performance.

3.3.1 Perception

Detail of the survey questions was based on initial anecdotal evidence from observing the course newsgroup and feedback from early tutorials. The same questionnaire was administered twice, once after the first assignment was completed (in week 7 out of 13), and again after the second assignment was completed (week 12). In addition to questions about the tutorials and quizzes, there were questions about the assignment, and about lectures. These additional questions provide a basis for triangulation (Cohen & Manion 1985). It was expected that student attitudes to the quizzes would evolve over the course of the experiment, since this was a relatively novel concept, while attitudes to assignments and lectures would not change much, as this was familiar ground. Consequently, any change in attitudes to tutorials, quizzes or peer assessment could be measured against change or lack of change in views on the less novel parts of the course.

The questions most relevant to this paper are listed here (using a 5-point Likert scale: 1 = strongly disagree, 5 = strongly agree):

- small quizzes after a tutorial make me work more consistently
- small quizzes after a tutorial make it easier to follow the next lecture
- small quizzes after a tutorial encourage me to keep up
- small quizzes after a tutorial are extra stress

- tutorials in this course are well-matched to lectures
- being given solutions to problems is more useful than attending a tutorial
- small quizzes after tutorials help me to understand the next tutorial
- marking another students work gives me extra insights
- marking another students work makes it clearer where I went wrong
- marking another students work has given me more insights for doing the assignment
- marking another students work has helped me understand what it takes to be an expert
- tutorials are linked to the assignment
- I am more prepared for tutorials than usual because of the quizzes
- lectures are more important than tutorials for understanding material
- it would be better to drop lectures and do more tutorials
- tutorials in this course are more useful than in other courses
- small quizzes after tutorials make it easier to keep up (compared with other courses)

These questions required short factual answers to be written in:

- My program is (mark one or fill in other): IT or Eng: EE CS SW other: . . . .
- My current GPA is (if known) . . . .   and in this course the grade I expect is . . . .

The following page contained open questions:

- The best feature of the tutorial quizzes is
- The worst feature of the tutorial quizzes is
- The effect of the approach in this course on your views on the usefulness of tutorials is

These questions were intended to uncover a spread of attitudes towards peer assessment and tutorials in general. While they do not provide a basis for comparison with any other way of running this course, the students in this class mostly have been at the university for at least 3 semesters, so their attitudes could be weighed to some extent against other courses they'd experienced.

3.3.2 Performance

Since perception can be incorrect, other measures of the effectiveness of the approach have been used as well. As noted before, too many things have changed to make a direct comparison with previous versions of the course convincing. However, such a comparison is at least some indication of the success of the approach. Given that similar ground was covered, and a similar style of final examination (multiple-choice), the students' overall results are some indicator of success.

Further, given that the intervention in tutorials was meant to foster deep learning, any evidence of stronger abilities to form theories and test them against reality would support the claim that the approach improved deep learning. The first of two assignments most closely fits these requirements. Students were given several sorting

algorithms, directly based on implementations in the pre-scribed book (Preiss 2000), but with names changed to force them to work out which was which. The algorithms were carefully selected so that they would exhibit different behaviours with different kinds of data:

- *quicksort* in two variations:
  - first pivot – the simplest version of quicksort, which has worst-case performance on data either already sorted, or in reverse order; in the worst case it requires time $O(n^2)$ and uses stack space $O(n)$ for recursion
  - median-of-3 pivot – an improved version of quicksort which is extremely unlikely to exhibit the worst-case behaviour (it almost always runs in time $O(n \log n)$, and uses $O(\log n)$ stack space)

- *merge sort* – more like the better quicksort in all cases, except is uses $O(n)$ extra memory, but not allocated on the stack, and so less likely to cause a run-time error message

- *insertion sort* – time $O(n^2)$ except if the data is already sorted, in which case, it becomes $O(n)$; no significant extra memory

The class was required to find the analysis (working out which algorithm was which and getting it from the book was acceptable), time the algorithms under a variety of conditions (ordered, random and reverse-order data, of varying sizes), and relate the measured times to the analysis.

This choice of assignment question is significant for a number of reasons:

- this style of problem has been called *empirical analysis*, and is known to be hard (Sanders 2002): students have difficulty relating theoretical results to measurements in the lab because to do so requires understanding what the theory signifies, an attribute of deep learning (Ramsden 1988)

- the algorithms were drawn from a section of the course not yet covered, so the ability to handle this material in the context of previously learnt theory showed the ability to relate new material to previous knowledge (Ramsden 1988)

- the kind of theory being applied was learnt through the tutorials and quizzes, and was the mathematical background to the course – traditionally the area students find most difficult not only to grasp but also to apply

The second assignment – implementing a simple text-based calculator (the arithmetic operations $+$, $-$, $\times$ and $\div$, with standard precedence and bracketing) – was not nearly as challenging from the point of view of developing original knowledge and understanding the theory.

Since the first assignment was most likely to test whether deep learning had taken place, it is useful as a basis of comparison to quiz performance. If the students did the quizzes consistently and did well in them, at least up to the date of this assignment, they could be expected to do better on the assignment than if they didn't do the quizzes. Since doing well on the assignment and doing well on the quizzes could be related by the common variable that good students will generally do well, it is useful to compare the correlation of quiz results with other assessments, to see if assignment 1 has a different outcome.

3.4 Conclusion

Given the limitations inherent in a study with no baseline and confounding variables, there is a limit to what can be read into results. However, correlating the first assignment and quiz results should provide some kind of measure of the effectiveness of the intervention.

Given that comfort factor has previously been identified as a significant success factor (Wilson & Shrock 2001), students' attitudes will be some kind of indicator of how successful the approach has been. Student perception, at least, is a measure of potential hostility to a new idea. However, measuring learning outcomes is also important, as student perception is not ultimately the variable of interest.

4 Results

Waste the time in tutorials $=>$ Tutes rushed ...basically means you should learn at home before coming to tutorials $=>$ Defeats the purpose. Marking is also pointless
– response to open question in first survey.

The introduction of peer-assessed quizzes was not popular – as exemplified by the opening quote of this section. Being forced to "learn at home"? What next? Being forced to think?

One of the key difficulties in educational research is separating perception from effect. While it is true that students have some idea of how well they have learnt something, it is also true that a general feeling of well-being can increase students' perception of learning. For example, studies of algorithm animators have shown that it is important to measure improved comprehension as well as student perception, as student perception can be favourable even when learning hasn't taken place (Wilson et al. 1996).

Results in this section are presented with an eye to gauging perceptions as one measure, since there was a limit to more objective measures, because too many variables could have effected outcomes.

One of the things to be learnt from this experiment is confirmation that student perception can be misleading, especially in an intervention designed to make them work more continuously. While others have reported on false positives in the sense of students being happy with something that did not work, students' negative perceptions in this case do not match measured learning outcomes. In particular, there is some evidence of a link between deep learning and the tutorial quizzes, even though they were unpopular.

The remainder of the section presents measurements in 4.1, followed by discussion in 4.2. The section closes with a summary of major findings.

4.1 Data

Data presented here falls into two major categories. Survey data represents student attitudes, and course results represent the effectiveness of the intervention (to the extent that its effects can be separated out).

Survey data includes analysis of questions on a survey taken twice at different points in the course. Most questions used a 5-point Likert scale; thematic analysis was used for open questions.

Student results are a combination of quiz results (to measure the direct utility of the quizzes), assignment results (to measure broader effectiveness of the course) and the final exam (to measure overall effectiveness of the course as a whole).

The remainder of this section presents each of these categories of data. Since the asssignment and examination results are relatively disjoint from the intervention re-
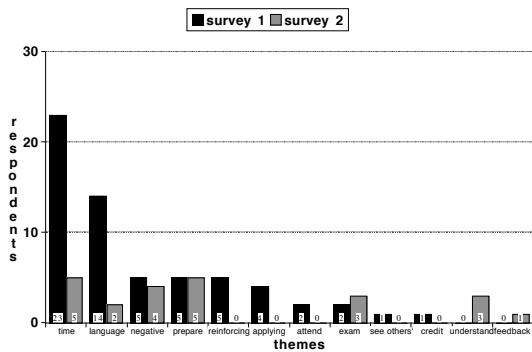
Figure 2: Thematic Analysis of Surveys.

(a) number of students out of 198 missing 0 to 11 quizzes

(b) fraction who submitted each quiz

Figure 3: Rate of submission of quizzes as indicator of tutorial participation.

ported on here, they are grouped together. Finally, relationships between assessment components are measured by correlations with quiz performance and attendance.
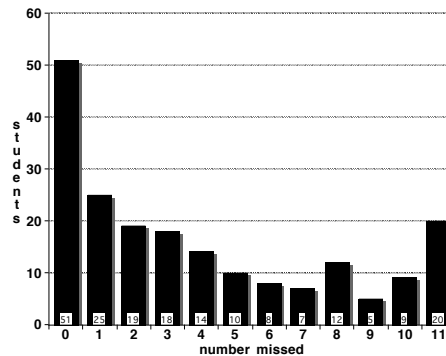
4.1.1 Survey results

The first survey was administered after the first assignment (15 September 2003, week 8 out of 13 semester weeks). The second survey was administered after the second (and last) assignment was completed (20 October, week 12). Figure 2 shows the progression of attitudes based on open questions as the course progressed. Response are ordered from most common to least common on survey 1. As can be seen, the strength of response on each point was not consistent across the surveys, indicating some change over the duration of the course.
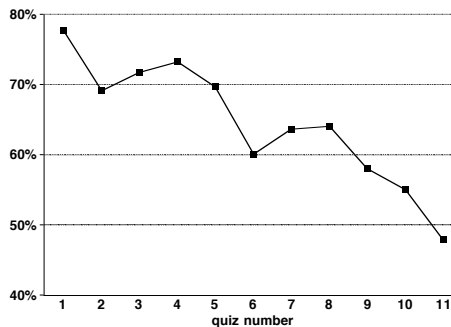
While there is a limit to what can be read into the change in open questions between the two surveys because a small fraction of the class wrote into the open question spaces, and a significant number of these expressed themselves in vague generalities which were hard to classify, the progression seen here makes some kind of sense.

Earlier in the course, time issues (the conflict between wanting more time for the quiz and more time for the tutorial) predominated ("time"), with complaints about the language of a tutor coming second ("language"). A group was simply negative, without being more specific ("negative"); a similar-sized group though acknowledged the value of the quizzes in ecouraging them to prepare in advance for tutorials ("prepare"). At this point, there is divergence between the first and second survey. In the first survey, the "reinforcing" effect of the quizzes was noted by a group, as was the value of "applying" knowledge. It appears that in the second survey, this group of responses had consolidated into a single group who acknowledged the value of the quizzes in aiding understanding. The value in seeing the work of others and credit for the quizzes were only acknowledged by a small group in the first survey, and none in the second. Some, it should be noted, also didn't like being given credit for quizzes, but this group was small and split into various cases – including no credit at all for quizzes or there should be more. Providing feedback was a new category in the second survey, but a small one.

From quiz 8 onwards, in response to the time issue, quizzes were moved a week later (rather than covering the ground of the tutorial, they covered the previous one). While the total time wasn't increased, it allowed more time for reflection and reinforcement. In practice, the value of this change was questionable: while there was a significant dip in attendance of quiz 6, most likely because it was in the last week of assignment 1, attendance increased for quizzes 7 and 8, but fell off sharply after quiz 8 (Figure 3(b)). This decline could have been because of outside pressures like end-of-semester accumulation of as-

signment load, but it does not provide strong evidence that the class was happier with this delayed quiz model.

Overall, migration of attitudes between the surveys is interesting and suggests that the value seen in quizzes increased, but too small a fraction filled in the open questions (about 50% in survey 2, but of these, a good fraction didn't answer all of the open questions) to draw strong inferences from this data.

Statistics in Figure 4 show that the tutorials and the peer-assessed quizzes were less popular than the lecture intervention (using motivating examples) and the assignments. While all categories improved between the two surveys, the tutorials improved more (12%, versus 7% improvement in mean score for assignments, and 4% for lectures). However, the tutorial average remains lower than the others: 2.9 versus 3.6 for both the other areas.

The effect of the poor English (as perceived by the class) of one tutor could be a significant factor, but on this data alone, it cannot be concluded that the intervention in tutorials was popular.

4.1.2 Quiz results

What's interesting about the quiz results is that anything which required algorithm analysis or design skills (as opposed to selecting between alternatives) tended to result in lower scores with a higher standard deviation. As can be seen in Figure 5, quizzes 1, 2 and 5 had lower averages than other quizzes, and a higher than usual standard deviation.

The increase in averages and increase in standard deviation for quizzes 8 and 9 suggests that the change of pace (quizzes a week after the related tutorials) suited students better on average, but some did worse through having a break of a week (2 weeks in the case of quiz 8 because of a mid-semester break). The decline in attendance in quizzes
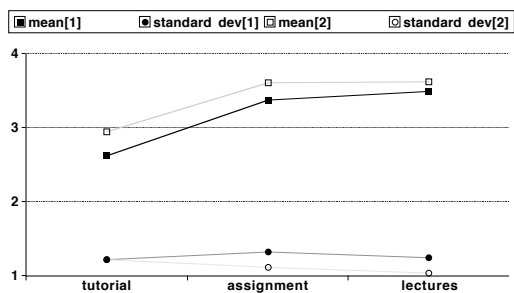
Figure 4: Means and Standard Deviations of Surveys 1 and 2. *Questions relating to tutorials (including peer-assessed quizzes) are separated from those relating to the assignment and lectures.*
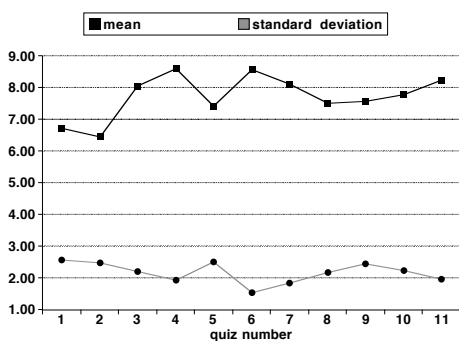

Figure 5: Quiz marks (average and standard deviation) out of 10 for 11 quizzes. *The first two counted 0.5% each; the remaining nine 1% each, for a total of 10% of the course.*

9 to 11 could also be a factor, so variation in students' results across the last few quizzes can't be put down to a single variable. Overall, however, it is hard to conclude that moving the quizzes a week later was advantageous.

4.1.3 Assignment and examination results

How did the class do in general? First of all, their overall result was quite good – almost 75% scored a grade of 5 or above. Grades at University of Queensland are on a 7-point scale, which translates to percentages as follows:

- 7 – 85% or more
- 6 – 75 to 84% or more
- 5 – 65 to 74% or more
- 4 – 50 to 64% (below 50% is a fail)
- 3 – 45-49%
- 2 – 20-44%
- 1 – below 40%

Figure 6 shows the grade distribution both as a percentage achieving a given grade, and as a percentage achieving that grade or better. For example, 90% achieved a grade of 4 (pass) or above. No one had a grade of 2, indicating that no one who made a serious effort scored a bad fail.

It is instructive to compare the 2003 grade distribution with those of previous years.

In Figure 7, it can be seen that there is a different distribution of final grades in previous years. In 2002, the language used was changed from Java to C++, which may have had some negative effects, since C++ is a more complex language, and Java was already known to the class from previous courses. The 2001 results are therefore more directly comparable to the 2003 results. Even so, the distribution is different in character. In 2001, by contrast with the equivalent 90% figure from 2003, 88% achieved a grade of 4 or better, which is similar. The tails of the distributions in 2003 and 2001 differ, however. 63% of the 2001 class achieved a grade of 5 or higher, versus 75% of the 2003 class. There were slightly more students in the 2001 class achieving a grade of 7 than a grade of 6, and more than twice as many achieved a 2 as achieved a grade of 3. The 2003 distribution, by contrast, is closer to a normal distribution shape, if slightly skewed to the high end. The 2002 distribution also deviates from a normal curve.

What these variations suggest is that both the 2001 and 2002 courses left some students behind, while being too easy for the best students. The 2003 result, with its more regularly-shaped distribution, appears to have been a better balance between challenging the good students and being accessible to the weaker students.

4.1.4 Comparison of Results

Figure 8 shows correlations between quiz scores and other assessments. Since attendance fell off after quiz 5, in addition to overall figures, the correlations are also broken down by the first 5 quizzes and subsequent quizzes. In addition to quiz scores, correlations with quiz attendance are also graphed.

What is interesting about these results is that quiz attendance (which tracks tutorial attendance) correlates relatively strongly with assignment results (0.55 for assignment 1; 0.49 for assignment 2), whereas there is a much lower correlation between quiz attendance and examination results (0.30). These figures suggest that attending the tutorials in itself was of most benefit for assignment 1, which was hypothesised as benefiting the most from deep learning. If quiz results alone are considered, however, the result is not as conclusive, as the correlations between the two assignments and quiz marks are very similar (0.50 for assignment 1; 0.51 for assignment 2). A further interesting point about the results is that the drop-off in quiz attendance after quiz 5 made a big difference to the correlations between quiz marks and other assessment components, but the correlation between quiz attendance and other assessment components was unchanged (except assignment 1, which was over by that time). This last measure suggest some validity in the correlations. Assignment 1 performance should not have been affected by quiz at-
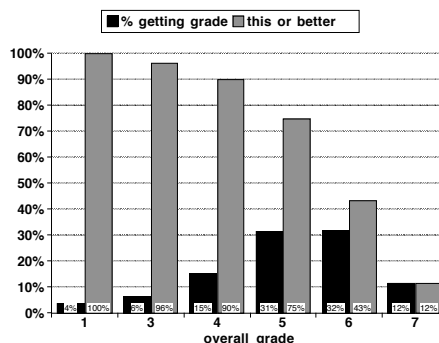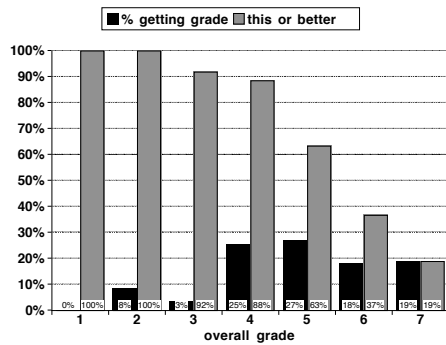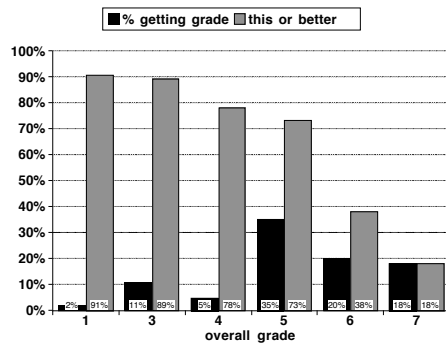

Figure 6: Grade distribution shown as percentage with each grade and percentage who achieved each grade or better.

(a) 2001

(b) 2002
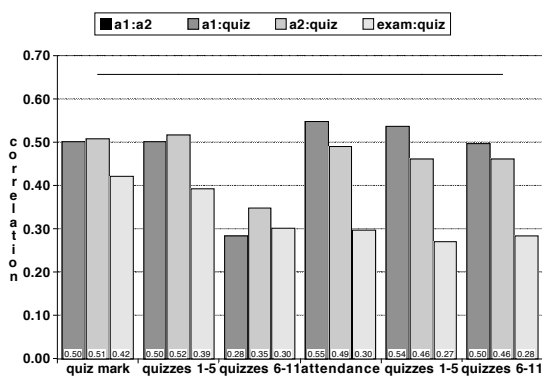
Figure 7: Grade distributions from previous years.

Figure 8: Correlations between quizzes and other assessment. *The first 3 data sets compare quiz results; the last 3, quiz attendance; the horizontal line is the correlation (0.66) between the two assignment results,* **a1** *and* **a2**.

tendance later in the course, but assignment 2 and examination performance could be. All of these correlations are significant at the 0.05 significance level for a sample size this big (over 200).

4.2 Discussion

Overall, the comparison of results using correlations suggests that enforced attendance at tutorials does have some beneficial effect on student learning. However, it is not so clear that the quizzes alone were beneficial because it is not true that assignment 1 correlates better with the quiz results than assignment 2. The strongest results suggesting that active learning was promoted was the correlation between quiz attendance and assignment 1 results, which

was the strongest correlation, other than that between assignment 1 and assignment 2.

The quiz results have a stronger relationship to the assignment results than to the final examination, suggesting that they did have some relationship to the style of learning being assessed in the assignments. Since the examination was multiple-choice, this result is not unexpected.

The final grade distribution improves on the two previous years: deviation from a normal distribution suggests some members of the class were left behind, while the course was too easy for others. However, the cause of this apparently successful outcome is hard to isolate.

4.3 Summary

Overall, the results do not strongly support the case that tutorial quizzes in themselves were a very useful innovation. Students did not like them, and the correlation results suggest that enforced tutorial attendance and possibly prior tutorial preparation were a stronger effect. If, however, the quizzes resulted in these effects, they had some value. Perhaps this can be seen as another example of assessment driving student behaviour (Gibbs 1999).

The fact that the class did better overall, with a more regular distribution of final grades, than previous classes suggests some value in the approach – even if it is hard to separate out from other changes.

5 Reflections

The use of peer assessment is growing in acceptance. However, its use in a course on data structures and algorithms appears to be novel. In Computer Science education, the idea has appeared in a few places, even if the approaches have not placed a strong emphasis on any specific learning model.

Action learning and the social construction model are not well known in Computer Science education research. Similar ideas can be found in moves in the last decade to reform engineering education, but those moves were based on pragmatic repair of what students did not like, rather than on a move away from existing models.

The nearest idea in current Computer Science thinking – and in related engineering disciplines –is the spiral model of software development (Boehm 1988), which is similar in broad terms to the action learning cycle. Given that there is a strong acceptance of the value of group work, and a growing understanding of the social nature of software development, Computer Science education is ready for a change to a stronger emphasis on the social construction model.

Despite the limited nature of the findings reported here, the results were interesting. Students' perceptions are important, because they effect performance – and they had prior experience of courses run in a different style. Academic performance provides another measure, if one which is hard to link to one change when there have been several changes. The fact that quiz attendance correlates strongly to assignment performance, especially the one most strongly assessing deep learning, suggests some value in the approach, though the students' negative perceptions suggest seeking other approaches aimed at the same effect. For example, quizzes could be held every second week, or at a lecture the week after each tutorial, to work around the practical problem of the short time available for tutorials.

Algorithmic thinking is hard, and the difficulty is compounded by the fact that more advanced concepts are layered on top of concepts learnt earlier in the course. This is a course where deep learning and reflection should make a very big difference to students' ability to cope.

The overall results suggest that some success was achieved; how this positive aspect can be implemented in a form more palatable to students remains a challenge.

References

Adams, W. A. (1980), *The Experience of Teaching and Learning: A Phenomenology of Education*, Psychological Press, Seattle.

Berger, P. L. (1966), *The social construction of reality : a treatise in the sociology of knowledge*, Doubleday, New York.

Biggs, J. (1999), 'What the student does: Teaching for enhanced learning', *Higher Education Research $ Development* **18**(1), 57–75.

Boehm, B. W. (1988), 'A spiral model of software development and enhancement', *Computer* **21**(5), 61–72.

Bradley, C. & Oliver, M. (2002), 'The evolution of pedagogic models for work-based learning within a virtual university', *Computers & Education* **38**(1–3), 37–52.

Brookes, W. & Indulska, J. (1996), Teaching internet literacy to a large and diverse audience, *in* 'Proceedings of the second Australasian conference on Computer science education', ACM Press, pp. 7–15.

Brookshear, J. G. (1985), The university computer science curriculum: education versus training, *in* 'Proceedings of the sixteenth SIGCSE technical symposium on Computer science education', ACM Press, New Orleans, Louisiana, United States, pp. 23–30.

Brown, S. (1999), Institutional strategies for assessment, *in* S. Brown & A. Glasner, eds, 'Assessment Matters in Higher Education', Society for Research into Higher Education and Open University Press, Buckingham, UK, pp. 3–13.

Bunning, C. (2001), Turning experience into learning, *in* L. Steffe & J. Gale, eds, 'Action learning at work', Gower, Aldershot, Hampshire.

Clanchy, M. (1993), *From memory to written record, England 1066-1307*, 2nd edn, Blackwell, Oxford.

Cohen, L. & Manion, L. (1985), *Research Methods in Education*, 2nd edn, Croom Helm, London.

Denning, P. J. (1999), Computing the profession, *in* 'The proceedings of the thirtieth SIGCSE technical symposium on Computer science education', ACM Press, pp. 1–2.

Director, S., Khosla, P., Rohrer, R. & Rutenbar, R. (1995), 'Reengineering the curriculum: design and analysis of a new undergraduate electrical and computer engineering degree at Carnegie Mellon University', *Proc. of the IEEE* **83**(9), 1246 –1269.

Gehringer, E. F. (2001), Electronic peer review and peer grading in computer-science courses, *in* 'Proceedings of the thirty second SIGCSE technical symposium on Computer Science Education', ACM Press, pp. 139–143.

Gibbs, G. (1999), Using assessment strategically to change the way students learn, *in* S. Brown & A. Glasner, eds, 'Assessment Matters in Higher Education', Society for Research into Higher Education and Open University Press, Buckingham, UK, pp. 41–53.

Hübscher-Younger, T. & Narayanan, N. (2003), Constructive and collaborative learning of algorithms, *in* 'Proc. SIGCSE Tech. Symp. on Computer Science Education', ACM Press, pp. 6–10.

Lave, J. & Wenger, E. (1991), Legitimate peripheral participation, *in* 'Situated Learning: Legitimate Peripheral Participation', Cambridge University Press, Cambridge, pp. 89–117.

Moore, J. W. (1998), 'Education versus training', *J. Chemical Education* **75**, 135.

Naps, T. L., Rling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S. & ngel Velzquez-Iturbide, J. (2003), 'Exploring the role of visualization and engagement in computer science education', *ACM SIGCSE Bulletin* **35**(2), 131–152.

Papert, S. (1993), *Mindstorms*, 2nd edn, Basic Books, New York.

Preiss, B. R. (2000), *Data Structures and Algorithms with Object-Oriented Design Patterns in Java*, Wiley, New York.

Ramsden, P. (1988), Studying learning: Improving teaching, *in* P. Ramsden, ed., 'Improving Learning: New Perspectives', Kogan Page, London, pp. 13–31.

Ruehr, F. & Orr, G. (2002), 'Interactive program demonstration as a form of student program assessment', *The Journal of Computing in Small Colleges* **18**(2), 65–78.

Sanders, I. (2002), Teaching empirical analysis of algorithms, *in* 'Proceedings of the 33rd SIGCSE technical symposium on Computer science education', Cincinnati, Kentucky, pp. 321–325.

Sanders, I. & Mueller, C. (2000), A fundamentals-based curriculum for first year computer science, *in* 'Proc. 31st SIGCSE Tech. Symp. on Computer science education', Austin, TX, pp. 227–231.

Schön, D. (1995), 'The new scholarship requires a new epistemology', *Change* **27**(6), 27–34.

US (2003), 'Economic indicators May 2003'. http://frwebgate.access.gpo. gov/cgi-bin/getdoc.cgi?dbname= economic_indicators&docid=00my03. txt.pdf.

Vat, K. H. (2000), Training e-commerce support personnel for enterprises through action learning, *in* 'Proc. 2000 ACM SIGCPR conf. on Computer personnel research', ACM Press, Chicago, Illinois, United States, pp. 39–44.

von Glasersfeld, E. (1995), Constructivist approach to teaching, *in* L. Steffe & J. Gale, eds, 'Constructivism in Education', Lawrence Erlbaum Associates, Hillsdale, NJ.

Wilson, B. C. & Shrock, S. (2001), Contributing to success in an introductory computer science course: a study of twelve factors, *in* 'Proc. 32nd SIGCSE Tech. Symp. on Computer Science Education', ACM Press, Charlotte, North Carolina, United States, pp. 184–188.

Wilson, J., Aiken, R. & Katz, I. (1996), Review of animation systems for algorithm understanding, *in* 'Proceedings of the 1st conference on Integrating technology into computer science education', ACM Press, pp. 75–77.