Uses of Peer Assessment in Database Teaching and Learning

James Paterson[1], John N. Wilson[2], and Petra Leimich[3]

[1] School of Engineering and Computing, Glasgow Caledonian University,
Glasgow, G4 0BA. UK. `James.Paterson@gcu.ac.uk`
[2] Department of Computer and Information Sciences, University of Strathclyde,
Glasgow, G1 1XQ. UK. `john.wilson@cis.strath.ac.uk`
[3] School of Computing and Engineering Systems, University of Abertay Dundee,
Dundee, DD1 1HG. UK. `p.leimich@abertay.ac.uk`

Abstract. This discussion paper introduces three very different methods and contexts for the use of peer assessment in introductory database classes, each of which is supported by different learning software tools. In the first case study, at Glasgow Caledonian University, Contributing Student Pedagogy is used, where students contribute to the learning of others through the collaborative creation of a bank of self-assessment questions. This is supported by the Peerwise software tool. Secondly, at the University of Strathclyde, students undertake formative assessment of others in providing feedback on an initial element of a larger coursework assessment. A number of virtual learning environments (VLEs) are capable of supporting this method through customisable discussion fora. Finally, at the University of Abertay Dundee, peer and self assessment are used in a group project to adjust the group grade for individual students. This is effected through the use of the WebPA software tool.

1   Introduction

Peer assessment by students of other students' work, both formative and summative, has many potential benefits to learning for all concerned. It develops the ability to evaluate and make judgements, and in doing so students gain insights into their own learning. The results of evaluation by peers can also provide a valuable source of feedback. This paper describes a variety of approaches to peer assessment and feedback, and the software tools which support them, which have been used within introductory database classes.

2   A Contributing Student Pedagogy in an Introductory Database Class

A Contributing Student Pedagogy (CSP) is an approach in which students contribute to the learning of others, and value the contribution of others [7]. One example of a software tool which provides support for the implementation of

a CSP is PeerWise [4]. PeerWise provides a means to create an online repository of multiple choice questions (MCQs) in which students themselves write the questions. Students can then answer questions which have been contributed by others, and they have the opportunity to evaluate those contributions. The authors of PeerWise assert that asking students to write MCQs, and to provide appropriate explanations, gives a richer and deeper learning experience than simply answering practice questions which have been provided by staff [3]. The possibility that questions may be poorly thought out, or that the provided answers may be wrong, gives students an opportunity to develop skills in critical evaluation. The PeerWise system is in use in a number of institutions throughout the world. The relationship of its use to exam performance and the topic coverage represented in the students' contributions in introductory programming courses has been studied [3,5]. Recently, PeerWise has been used in an introductory database course at Glasgow Caledonian University. This section of the paper reports on the experience and on the topic coverage represented in the student contributions within that course.

2.1 Implementation

The module, Introduction to Database Development, was delivered over a short (6 week) timescale as part of a set of short introductory modules within a first year course which is common to all computing programmes. The main assessment instrument is a hand-in assignment, but there is also an online MCQ test. To encourage participation in PeerWise, a component of the overall module mark (10%) was awarded on the basis of that participation. Participation was required to be completed within weeks 2 to 5 of the module. To attain full credit, students were required to contribute at least 5 questions and answer 10 questions contributed by others. Students were made aware that they were not being assessed on the quality of their questions or the correctness of their answers. For each question, students were required to provide a question stem and a set of up to five answers, and also to indicate which answer they consider to be correct. They could optionally provide an explanation of their answer. On answering a question, a student can see the question author's choice of correct answer, and also the distribution of answers previously given for that question. It is entirely possible that the indicated correct answer may not in fact be correct, and the weight of opinion expressed in other students' answers may reflect this. The student can then optionally rate the question on a scale of 0 to 5 and provide a text comment. Factors which students may take into account in rating questions may include, for example, the correctness of the given answer and the quality of the explanation.

A total of 105 students contributed questions, which essentially is all the students who engaged with the module. Of these, only 4 contributed less than the required 5 questions. The highest number of questions contributed by any student was 10. The majority of students contributed exactly 5 questions. The total number of questions submitted was 545, and the average number of responses to each question was 2.8. Most students answered 10 questions, or a few

more than that. However, 15 students answered double the required amount of questions or more, and the highest number answered by any student was 45.

2.2 Evaluation

Evaluation of the CSP approach has focused initially on two aspects. Question quality is likely to be an indicator of depth of learning. Writing a question which is challenging, and to provide good explanations for correct and incorrect choices of response requires a good understanding, as does recognizing a good question when providing ratings. Topic coverage gives a collective view of the students' viewpoint on the course material and the relative importance of each topic.

Question quality Denny et al. [6] have applied a metric to measure objectively the quality of student-created questions in their courses. This has not been done yet in the initial analysis described here which focuses on the student ratings. The average rating of questions which were rated by more than 10 respondents (a figure chosen to provide a reasonable 'body of opinion') was 3.3. It is interesting to consider what students consider to be a 'good' question. For example, there was relatively little difference in average rating between a question which was a simple true/false question (True or False: SQL can be used to create a database?) and a more sophisticated question which requires a set of four SQL statements to be examined to identify the one which would correctly produce a specified result. It seems likely that experienced instructors would rate the questions significantly differently from the students. However, it is evident that students give low ratings to questions which they consider to have errors, for example where the stated correct answer is not correct, or where more than one answer could be correct. The provision of an explanation along with a question appears to have little influence on the rating. From the questions with more than 10 ratings, the average rating for questions with explanations was 3.3 compared to 3.1 for those without. In fact, less than 25% of the questions included explanations. It should be noted that no guidance was given to students on what constitutes a 'good' question. Given that the aim of the exercise is to enrich the learning experience, not to simply create a bank of practice questions, the quality of the questions is not the main concern. However, it seems likely that providing some discussion of question types and quality could be beneficial in encouraging students to devise questions which require deep understanding and to provide explanations.

Topic coverage Denny et al. [4] studied the coverage of topics within student-created MCQs in an introductory programming course. They concluded that the coverage was broad, included all major topics and showed a distribution similar to that in assessments set by instructors. They also noted that a significant number of questions touched on 2 or more topics. A preliminary analysis along similar lines has been done here with the database questions. One of the decisions which has to be made in this analysis is the granularity with which topics are
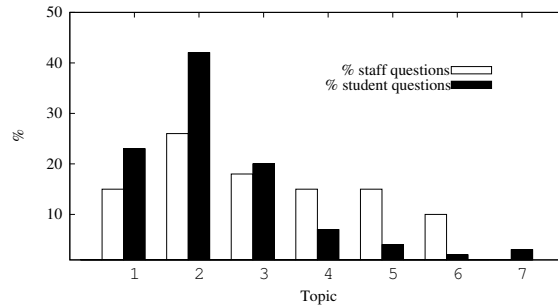
Fig. 1. Topic coverage of student and staff questions. Numbers on horizontal axis refer to course material chapters, the content of which is indicated in the text

defined. Denny et al. used coursebook chapters as topic descriptors, and noted in support of this decision that these topics matched well to those identified in an international study of introductory programming courses involving 200 teachers at University level [12]. We have used a similar classification according to the course structure. The course materials were organised into 6 chapters:

1. Introduction to database systems, nomenclature and data models
2. Database design and relationships
3. SQL SELECT and aggregates
4. Normalisation
5. Indexes, SQL JOIN, INSERT, DELETE, UPDATE
6. Building database applications

This is a preliminary classification, and is not necessarily ideal. In particular, there are in some cases several identifiable topics within a particular chapter. On the other hand, the structure is used by staff to guide the setting of MCQ assessments, and is familiar to students.

Figure 1 shows the distribution among these topics of students-contributed questions compared to that of staff-created assessment questions. Topic 7 is included to cover topics which are not explicitly included in the course but which students may have included on topics discovered through independent reading. Figure 1 shows that all major topics have indeed been covered. The distributions of student and staff questions are quite different. Although both show a bias towards topic 2 (database design and relationships), the bias is more marked in student questions, while coverage of topics which are delivered later in the module is very limited. It was also noted that questions were in almost all cases focused on isolated topics, with no real integration of topics within a single question. Interestingly, the latter comment also applies to the staff questions. There may be several reasons behind these results. The timescale of the exercise was quite different from that described by Denny et al. These questions were contributed within the teaching time on a short module, rather than over a period including exam revision time after teaching, which may explain a bias towards

topics taught early in the course. Students essentially had 4 weeks to contribute questions on topic 2, and less than 2 weeks for topic 5. The nature of the topics covered in a database course may lend themselves differently to creating questions than the topics in a programming course. Further work is required on topic categorisation within database courses, and on the use of PeerWise in courses with different structures and timescales.

2.3   Conclusion

The PeerWise system has been used in an introductory database course to support the creation by students of a bank of MCQs. Preliminary work has been done in analysing the contributions to investigate students' perceptions of question quality and their preference for topics on which to devise questions. For the latter, there is some evidence of differences from the picture observed in programming courses, but further study is required to clarify this. Further study is also required to evaluate the contribution to the students' learning experience.

3   Peer-based feedback

The issue of feedback is consistently highlighted in the context of student opinions of their experiences in higher education. Typically universities and colleges have moved to encourage quick and constructive feedback from lecturers to their students. There is however still an issue of the need to provide intermediate feedback on students' ongoing work and it is here especially that peer-based feedback can make a particularly useful contribution. As well as helping the recipient of the feedback, students also benefit by viewing the work of their colleagues from a critical perspective. Students find that peer review based assessment encourages reflective thinking and self-improvement at the expense of additional time demands. While there are issues over the integrity of this process, simple, holistic feedback on review material submitted provides sufficient student benefit to make this approach worthwhile [10]. It is further recognised that peer based review can make a large contribution to building scholarly communities [1] as well as leading to significant improvements in the quality of student work [13].

Arranging that these benefits are available in the setting of large class cohorts is a major problem that was addressed by the Open University in the context of its introductory programming class by using a proprietary virtual learning environment. The approach developed requires four discrete stages:

1. Registration
2. First posting - bulletin boards are write only
3. Follow up posting - bulletin boards are read/write
4. Work completion - bulletin boards are read only

The stages govern the permissions on the board and this fits with the work to be carried out by the student. In stage 2, students are expected to post a part of their initial solution. In stage 3, they are expected to review the work of another

student and write some helpful comments. In stage 4, they complete their own work using the insight gained during stage 3 both from their own interaction and by viewing the interactions of their peers. This approach is particularly suited to complex problems where there are a number of ways of addressing the issues involved. Entity-relationship modelling presents a very apposite domain for this kind of interaction. It is a social process where the designer can learn a lot from iterations in the solution space. This is enhanced by discussing the problem with others and considering alternative and sometimes incorrect viewpoints.

This study was carried out on an introductory Databases class for second level students. The class ran in a single semester and provided 10 credits. The cohort included 107 students, most of whom were registered on a BSc Honours degree in Computer Science. Students were presented with a scenario based on a hospital, which could be represented in twelve entities and eleven relationships with assorted degree and optionality. By this stage in the module, the concept of supertype/subtype entities had been introduced and there was a clear opportunity to use such a formalism in the employment hierarchy contained within the scenario. Some many-to-many relationships typically emerged during the analysis and there existed the potential for a recursive relationship to represent supervision. In all, the problem represented a variety of challenges to students and had a number of elements that were not straightforward to represent. The overall problem could generate a number of correct solutions.

A number of common software platforms are capable of supporting the previously described posting sequence. These can be categorized as virtual learning environments (VLE) or generic software that can be tailored to incorporate the necessary capabilities. The Blackboard[4] VLE provides a typical example of the former category. It supports a discussion board idiom that includes the basic functionality that is needed to control asynchronous group interaction. The system allows the instructor to manipulate fora and allocate users in particular groups to access these fora. The bulletin board supports text as well as paste-in images and attachments. Control of permissions is mainly to define post anonymity, removal and tagging of posts, thread creation and moderation. Moodle supports a similar group-based forum concept. This implementation controls the visibility of forum contents to users from outside the forum. It restricts students to posting to one thread but permits reply postings to be made to multiple threads. Moodle[5] allows pictures to be inserted directly into postings. Turnitin[6] has also been extended to provide equivalent functionality.

WebBBS[7] presents a typical example of the range of systems available in the second category of software platforms. It can be tailored to support the required permission pattern to structure peer-based feedback in an appropriate way. The interface is very simple with postings being categorised into threads. Images can be included by writing html tags into the test of the posting.

---

[4] http://www.blackboard.com/

[5] http://moodle.org

[6] http://www.submit.ac.uk/static_jisc/ac_uk_index.html

[7] http://awsd.com/scripts/webbbs/

*** Welcome ***	John Wilson	Mon, 19 Oct 2009, 6:29 p.m.
phase2_DBClasswork_200988029	student1	Thu, 22 Oct 2009, 11:30 a.m.
Re: phase2_DBClasswork_200988029	student2	Mon, 26 Oct 2009, 10:54 a.m.
Re: phase2_DBClasswork_200988029	student1	Tue, 27 Oct 2009, 2:20 p.m.
Nurses Model	student2	Fri, 23 Oct 2009, 2:59 p.m.
Re: Nurses Model	student3	Wed, 28 Oct 2009, 12:43 p.m.
CW1 Phase 2 Post	student3	Fri, 23 Oct 2009, 4:08 p.m.
CW1 Partial Solution	student5	Fri, 23 Oct 2009, 4:09 p.m.
Re: CW1 Partial Solution	student5	Tue, 27 Oct 2009, 2:54 p.m.
Re: CW1 Partial Solution	student4	Wed, 28 Oct 2009, 3:43 p.m.
Patient Solution	student4	Fri, 23 Oct 2009, 4:55 p.m.
Re: Patient Solution	student1	Sun, 25 Oct 2009, 6:04 p.m.
Re: Patient Solution	student4	Tue, 27 Oct 2009, 11:08 p.m.
Re: Patient Solution	student1	Wed, 28 Oct 2009, 3:46 p.m.
*** End of Phase 2 ***	John Wilson	Sun, 25 Oct 2009, 4:58 p.m.
*** End of Phase 3 ***	John Wilson	Wed, 28 Oct 2009, 6:05 p.m.

Fig. 2. Typical pattern of postings in one bulletin board group.

WebBBS was chosen for the forum structure because of its simplicity and adaptability. Control of the posting sequence was implemented as part of the WebBBS Perl script. Students were allocated to groups of four or five and provided with instructions on the overall objectives of the process, the details of how to make postings and the various deadlines. Figure 2 illustrates the typical sequence of interactions between users of the system. A good posting will include a description of a difficult part of the problem and comments such as:

> In the above diagram I have produced the relationship treat between patient and doctor. But in the scenario it says that a patient is assigned to a consultant. What I am unsure of is whether my diagram should be changed so that the relationship is between patient and consultant

A poorer solution would simply list all the entities and relationships in the scenario without an attempt to draw out the difficult elements.

3.1 Conclusion

This approach to organizing peer review has been found to be very effective in motivating student involvement in assessed coursework well in advance of the final deadline. The imposition of intermediate deadlines ensures that work is carried out in a more even manner than is typically the case with a single deadline. Most students take part in the process and make contributions during both the posting phases. The overall quality of the postings is variable, with some students misinterpreting the instructions and posting their complete solution or making only a very terse contribution. Similarly response postings vary considerably in quality. Students are instructed that the tone of the response postings is to be constructively critical and this always maintained. The postings are not anonymous and this may produce a constraint on the material that is posted. Whilst WebBBS provides a useful platform for this kind on interaction, the approach to posting images can present technical challenges to students at this level.

4 Peer assessment in the allocation of grades for group projects

The introductory database module at Abertay is taken in semester 2 of the first year by students from a number of computing-related courses. The class contact comprises a one hour lecture and two 1.5-hour labs per week per student, over 12 weeks. Group projects are used as the major assessment. These have been successful in enhancing and sustaining students' interest in the subject matter, and working closely with peers generally helps students undertake more active enquiry, contributing to the group effort. Also, reserving part of the weekly lab sessions for project work enables a close dialogue between the students and lecturer. The final group product is graded, and forms the majority of the module grade. As with group projects in general, students are concerned that grades should be fair, i.e. vary depending on the contributions made by individuals. This is mirrored by staff concerns that students should not pass the module purely on the strength of others. Group presentations, where students are graded individually, contribute to this but are insufficient as the weighting is fairly low compared to the product itself. Therefore the grading of the product (the group's database application and associated documentation) also required a way of individually adjusting grades. Initially, an attendance-based method was used. Students were allowed to miss two of the weekly in-lab group meetings without penalty, to account for unavoidable illnesses etc. However, the grades of students who missed more than two meetings were reduced by one number grade (on a 20 point scale where 20 represents A+, 9 represents D-, 0 non-submission) per meeting missed. This method was found to address the issue of contribution, but to have some major drawbacks: firstly, it relies on the lecturer maintaining accurate attendance records at all times, secondly, and more importantly, it also penalised students who made their contribution to the group work mainly outside of class, while not penalising students who contributed little despite attending class. Finally, this method had no student input, which is now regarded as good practice in assessment and evaluation (see, e.g. [11]). In an attempt to improve the process, and to avoid the drawbacks described above, peer assessment as part of the grading process was first introduced last year.

4.1 Implementation

In researching the experiences of others, several themes emerged. Several authors have described problems where students were required to discuss and agree their grades publicly within the group. For example, Cogdell et al [2] reported that students "did not like giving low marks to colleagues face to face. Consequently non-contributors would get the same marks as everyone else and the rest of the group would feel resentful. Alternatively the group would mark a member down and this person would complain vociferously." Other authors agree that peer assessment should be performed in private; for example, Lejk and Wyvill [9] found that students were more discriminating in their peer assessment when it is

performed secretly. Lejk and Wyvill [9], among others, also emphasise the importance of including self-assessment in the process, in order to avoid over-generous students effectively penalising themselves. Kennedy [8] presents another, similar scenario, and found that there was little overall variation introduced in the grades through the peer assessment process, and that many students expressed reluctance to judge their peers. On the other hand, Kennedy observed that other students were keen to discriminate, and that this could lead to dysfunctional groups, with uneven distributions of tasks in the group right from the beginning of a project, when domineering students ensured they undertook the most demanding and credit bearing tasks. Kennedy questions the reliability and validity of the process also because of observed wide inconsistencies in students' judgment of each other. In Kennedy's scenario, self assessment was not incorporated, and whether the peer assessment was public or confidential is not stated.

Based on the literature, it was decided that the peer assessment used should be confidential, and include self-assessment. WebPA [14, 15], open source software developed for this setting by the University of Loughborough, was used to facilitate the process and minimise any administrative burden. The students in each group were asked to rate every group member in five distinct areas, based on those suggested by WebPA [14]:

1. Co-Operation: This covers attendance at meetings, contribution to meetings, carrying out of designated tasks, dealing with problems,. helping others in the group.
2. Communication: This covers effectiveness in meetings, clarity of work submitted to the group, negotiation with the group, communication between meetings and providing feedback about the contributions made by others in the group.
3. Enthusiasm: This covers motivation, creativity and initiative during the project, including finding out about methods beyond the taught materials.
4. Organisation: This covers skills in self-organisation and the ability to organise others. It also covers planning, setting targets, establishing ground rules and keeping to deadlines.
5. Contribution: This covers the overall effort put in by an individual during the Semester.

For each area, a rating scale of 0-5 was used:

Score 0 : no help at all
Score 1 : quite poor
Score 2 : not as good as most of the group
Score 3 : about average for this group
Score 4 : better than most of the group
Score 5 : really excellent.

The WebPA software allows staff to review individual results and also aggregates the scores into a final grade. Staff also select the weight of the peer assessment component, in this case 50%. An example of the WebPA output for a single project group, is shown in Figure 3. The final grades are on the Abertay

Results for: Student A	From Student: A	B	C	D	E
1. CO–OPERATION	4	5	4	3	3
2. COMMUNICATION	4	3	5	2	1
3. ENTHUSIASM	4	4	4	3	3
4. ORGANISATION	3	3	3	2	2
5. CONTRIBUTION	4	5	4	1	2

Results for: Student B	From Student: A	B	C	D	E
1. CO–OPERATION	5	4	5	3	2
2. COMMUNICATION	5	4	5	1	1
3. ENTHUSIASM	5	5	5	3	3
4. ORGANISATION	4	4	4	4	1
5. CONTRIBUTION	5	5	5	5	5

Results for: Student C	From Student: A	B	C	D	E
1. CO–OPERATION	4	5	3	3	0
2. COMMUNICATION	4	4	4	2	0
3. ENTHUSIASM	4	4	3	3	0
4. ORGANISATION	3	5	3	2	2
5. CONTRIBUTION	4	5	3	1	0

Results for: Student D	From Student: A	B	C	D	E
1. CO–OPERATION	0	0	1	3	2
2. COMMUNICATION	0	0	2	3	1
3. ENTHUSIASM	0	0	1	3	2
4. ORGANISATION	0	0	0	3	2
5. CONTRIBUTION	1	0	1	3	3

Results for: Student E	From Student: A	B	C	D	E
1. CO–OPERATION	2	1	3	3	2
2. COMMUNICATIO	1	1	4	4	2
3. ENTHUSIASM	2	2	4	3	4
4. ORGANISATION	0	1	3	4	2
5. CONTRIBUTION	2	2	4	5	4

Overall group mark Name	80% Web–PA score	B16 Final Grade	
Student A	1.16	86.52	B17
Student B	1.4	95.99	A19
Student C	0.98	79.06	B16
Student D	0.5	60.11	C12
Student E	0.96	78.32	B16

Fig. 3. Sample WebPA output for one project group

grading scale (where A20 is the best possible; D9 the lowest pass mark). The group mark, B16, was converted to a percentage and then the individual grades converted back to the Abertay scale.

4.2 Evaluation

Once installed by Information Services, WebPA was easy to set up. This requires a number of steps, uploading module and student information (using templates provided), creating the assessment with relevant dates and a marking form, assigning the students to groups. WebPA allows for the use of single sign-on. In the second year of operation however, unforeseen problems were encountered in that most students could not log in and received no error message either. Eventually, the problem was fixed by Information Services. It turned out that authentication relied on the students' email address, rather than their user name, and due to very recent migration of the student email provider, email addresses did not all follow a single pattern. Despite these teething problems, which resulted in a two-week delay and countless unsuccessful attempts, students were keen to participate. However, it was decided not to apply the 100% non-submission penalty for students who had clearly participated in the project and earlier plans to seek formal feedback on the process from students this session were abandoned.

Once the students had left their peer and self assessments, group grades were entered into WebPA and its algorithms applied to calculate individual grades. The results, and detailed ratings, were inspected closely by the lecturer. As expected from the observations of group work in labs, several patterns of group outcomes emerged. In some groups, there was very little (or even no) variance, resulting in all students being awarded the group grade. In other groups, there were large differences, resulting in a wide differentiation of individual grades.

As in the example of Figure 3 above, the five different categories were used effectively, and different scores given. This indicates that students took care in arriving at their assessments. Many added thoughtful and reflective comments to

their scores. This helped reconcile the very occasional wide differences between the scores given to one individual by the group members. The written comments were also useful in the few cases where the ratings were very different from the lecturer's personal impression formed during classes. Where necessary, academic judgment and performance in other module elements were used to arrive at a final grade. There was no evidence of students "ganging up" on a group member, or of students trying to improve their grade through giving themselves a very high score.

Informal feedback was received from several students by email. This showed that students found the software easy to use. One student was initially concerned:

> Can I just confirm that my ratings for the other group members will not be displayed for all to see? I just feel that this will cause problems as already there have been accusations of 'Back stabbing' between group members, it's crazy I know.

Following assurance that individual ratings were confidential, the student commented:

> It worked fine and was very easy to use! I think all group work modules should have this at the end, it's excellent.

4.3   Conclusion

While a full evaluation is yet to be carried out, initial experiences with this method based on two instances of operation are positive. The system has now been used with two cohorts, each with about 20 groups and 80 to 100 students. Not a single complaint has been received about the system's use or the resulting grade, indicating that students do find the system fair.

5   Overall conclusion

The three very different examples introduced in this paper illustrate the potential for the use of peer assessment and feedback in introductory database classes in a wide variety of contexts and illustrate a range of software tools which are available to facilitate these processes. The approaches and tools used have been shown to be applicable in the teaching and learning of databases and have succeeded in engaging students in peer activity. Further work is required to evaluate the impact on learning and to identify activities and topics within database classes which may benefit from these approaches. As an aside, while the use of these tools is not specific to databases, or any other subject for that matter, they may have one additional use which is unique within our discipline: one thing they all have in common is that they each make use of a database and they could be nice examples of case studies which demonstrate the real-world importance of databases!

6   Acknowledgments

References

1. C. Chang, G. Chen, and L. Li. Constructing a community of practice to improve coursework activity. *Computers & Education*, 50(1):235–247, 2008.
2. B. Cogdell, A. Brown, and A. Campbell. Peer-assessment of group work in a large class - development of a staff and student friendly system, 2004. Available at ftp://www.bioscience.heacademy.ac.uk/TeachingGuides/fulltext.pdf.
3. P. Denny, J. Hamer, and A. Luxton-Reilly, and H. Purchase. PeerWise: students sharing their multiple choice questions. In *Proc. 4th International Workshop on Computing Education Research*, pages 51–58, 2008.
4. P. Denny, A. Luxton-Reilly, and J. Hamer. The PeerWise system of student contributed assessment questions. In *Proc. 10th Conference on Australasian Computing Education*, pages 69–74, 2008.
5. P. Denny, A. Luxton-Reilly, J. Hamer, and H. Purchase. Coverage of course topics in a student generated MCQ repository. In *Proc. 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 11–15, 2009.
6. P. Denny, A. Luxton-Reilly, and B. Simon. Quality of student contributed questions using PeerWise. In *Proc. 11th Australasian Computing Education Conference*, pages 45–53, 2009.
7. J. Hamer. Some experiences with the 'contributing student approach'. In *Proc. 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 68–72, 2006.
8. G.J. Kennedy. Peer-assessment in group projects: is it worth it? In *Proc. 7th Australasian Computing Education Conference*, pages 59–65, 2005.
9. M. Lejk and M. Wyvill. The effect of the inclusion of self assessment with peer assessment of contributions to a group project. *Assessment and Evaluation in Higher Education*, 26(6):551–561, 2001.
10. S. Lin, Liu. E., and S. Yuan. Web-based peer assessment: feedback for students with various thinking-styles. *J. Computer Assisted Learning*, 17(4):420–432, 2001.
11. D Nicol and D. Macfarlane-Dick. Formative assessment and self-regulated learning. *Studies in Higher Education*, 31(2):199–218, 2006.
12. C. Schulte and J. Bennedsen. What do teachers teach in introductory programming? In *Proc. 2nd International Workshop on Computing Education Research*, pages 17–28, 2006.
13. Y. Sung, K. Chang, Chiou S., and H. Hou. The design and application of a web-based self- and peer-assessment system. *Computers & Education*, 45(2):187–202, 2005.
14. J. Tenant, A. Crawford, and N. Wilkinson. Supplementary information about self and peer assessment. Available at http://staffcentral.bton.ac.uk/.
15. Loughborough University. WebPA. Available at http://www.webpaproject.com/.