EXPERIENCES WITH WEB-BASED PEER ASSESSMENT OF COURSEWORK

Hans Hüttel and Kurt Nørmark
Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, Aalborg, Denmark

Keywords: Peer Assessment.

Abstract: We describe experiments with a web-based system for peer assessment in a course on automata theory and program semantics. The paper describes the web-based assessment system, and experiences from the first round of use in 2011. We find a correlation between a high level of peer reviewing activity and a high grade obtained at written exam.

1 INTRODUCTION

In this paper we describe an experimental system whose aim is to address some of the challenges that we have found when giving undergraduate courses.

1.1 Challenges to University Teaching

The proportion of school leavers that enter higher education is steadily increasing, and this has led to the emergence of a larger group of students with different attitudes towards learning.

Biggs and Tang (Biggs and Tang, 2007) distinguish between the dedicated student –'Susan' – and the less dedicated student – 'Robert'. 'Susan' is interested in her subject and what she learns is important to her, while 'Robert' is not so interested in his subject, uses notably less effort and focuses on how to be able to qualify for a job. Traditionally, university teaching has focused exclusively on 'Susan'; an important challenge has been to cater also to the needs of 'Robert' while keeping the focus on the needs of the subject and of learning.

Observations suggests that whatever has been taught in a programme tends to be forgotten by students relatively soon. A survey by Anderson et al. (Anderson et al., 1998) documents that a majority of undergraduate-level mathematics were able to recall only little or none of what was taught in their first-year courses.

Our own experience is similar. We come across students at masters level asking us for help because they need to brush up their knowledge of central topics from courses that they passed earlier.

Another challenge is that, because of the increasing student intake, lecturers need to spend more time evaluating students but will not have more resources at their disposal.

Lauvås and Jakobsen (Lauvås and Jakobsen, 2002) point out both of these problems and argue that it is due to the fact that 'Robert' will choose an instrumentalist approach to learning whose primary goal is that of being able to pass the exam. 'Robert' will not prepare for course sessions but tries to cram the material during the last few days before the exam. This is a hindrance to deep learning.

The challenge is then to teach such that some 'Robert' students will end up becoming 'Susans' and in such a way that learning is improved, that deep learning is encouraged and in such a way that the teacher does not use more resources in this process.

1.2 Formative Evaluation through Peer Assessment

Peer assessment is a strategy that has received a great deal attention over the past 15 years in the world of education research. For a recent overview, see the survey paper by Topping (Topping, 2009). It has often often been argued that peer assessment can improve the reflective processes of learning, since students through the assessment process will have to reflect on how their fellow students are approaching the task of learning.

One often distinguishes between

- *summative evaluation*, which is the assessment whose aim is to determine if the learning goals

Hüttel H. and Nørmark K..
EXPERIENCES WITH WEB-BASED PEER ASSESSMENT OF COURSEWORK.
DOI: 10.5220/0003918401130118
In *Proceedings of the 4th International Conference on Computer Supported Education* (CSEDU-2012), pages 113-118
ISBN: 978-989-8565-07-5

of a teaching activity have been reached, and

- *formative evaluation*, which is the assessment whose aim is to guide the students so they will know how their learning is proceeding and what they should do to improve it. It is through formative evaluation that reflection can happen.

Often summative evaluation will take the form of a formal examination. On the other hand, formative evaluation will be part of the learning process. One can therefore seek to incorporate a notion of formative evaluation as part of teaching activities, and this is where peer assessment comes in.

The claims that Lauvås and Jakobsen make are (Lauvås and Jakobsen, 2002): If formative evaluation is thorough and if one insists that the students make a substantial effort throughout the course, the students will learn much more and will remember the material much better and for a longer period of time. The exam itself will become a different experience for students, since they will be examined on the basis of material with which they are already well acquainted and feel that they master. Lauvås uses the analogy of athletes that practice regularly and are told by their coach how to improve their performance.

Lauvås and Maugesten have experimented with methods for restructuring teaching via *peer assessment* (Maugesten and Lauvås, 2004). The activities have been laid out such that all students would explicitly need to adopt a 'Susan'-style learning strategy.

1.3 Existing Solutions

There has already been a fair amount of work on automating the administrative aspect of the peer assessment strategy.

Michael de Raadt has created a peer review module (de Raadt,) that can be used with Moodle (Moodle,). One limitation of this module is that peer reviews take the form of a series of multiple-choice questions for which boxes are to be ticked. Other forms of assessment are not possible.

In a series of papers, Joy and Sitthiworachart have focused on how to use web-based peer assessment to assist deep learning in programming classes (Sitthiworachart and Joy, 2004; Ward et al., 2004). In this work, they have among other things, focused on asking the students to comment on specific aspects of solutions to programming exercises. Their results indicate that web-based peer assessment can indeed assist in the process of deep learning.

1.4 Our Setting

We have developed and used an experimental system for peer assessment used in the course *Syntax and Semantics* taught at our university. Our goal has been to gain experience in how to increase student efforts and improve learning through web-based methods for peer assessment. We have based the design of our system on previous, largely paper-based strategies for peer assessment.

2 PREVIOUS EXPERIENCE

From 2004 to 2007, the first author has experimented with methods of peer assessment in another theory course on computability and complexity theory.

A questionnaire-based survey has shown that, in the past, a student would spend only 30 minutes preparing for a course session. While peer assessment dramatically increased student efforts, the solution was entirely paper-based and this required a significant effort of all involved. The incentive for students was that the participating students would have a reduced syllabus at the oral exam.

The assessment expected of students involved that of reviewing the solutions to textbook-style problems solved by their fellow students. Here, the problem turned out to be one of greatly varying levels of understanding: students that had difficulting solving a problem were often unable to provide meaningful reviews. Quite often, our conversations with students during commenting sessions revealed that the student had not read the relevant pages of the course textbook.

Finally, some 'Robert'-students very consistently handed in 'non-solutions' in which they simply wrote that they could not solve the problems that had been stated. These same students would also write very brief reviews of solutions. As they had handed in solutions and participated in the commenting process and thereby followed the formal procedures, it was hard to argue that they should not have a reduced exam syllabus even if they had done little actual work.

3 THE CURRENT COURSE

The course *Syntax and Semantics* is found at the 4th semester of the undergraduate programmes in Computer Science and in Software Technology. The course covers formal language theory and programming language theory.

The course consists of 15 4-hour sessions, divided into 15 practical sessions of 2 hours each and a 90 minute lecture. At the same semester, there were three other courses of 3 ECTS each and an 18 ECTS student project. As the students collaborate on projects in groups of 5 to 7 students and are assigned a group room, it is common practice for them to also work on course practicals in these project groups.

For the Computer Science students, the exam in Syntax and Semantics is a 3-hour written exam. Students following the Software Technology programme did not participate in the written exam. These student had the course assessed as part of their semester project which involves several other topics not related to the course syllabus.

4 THE APPROACH

The course consisted of a variety of teaching activities, includes lectures and problem solving. However, in this section we concentrate on the new development, namely the approach that we have followed for peer assessment, and how it led to the design of our system. First, the overall approach to questioning, answering, and reviewing is described. This includes the means of incentives provided by the approach. Next, the workflow of a single session - called an *answering session* - is explained.

4.1 The Peer Assessment Approach

Our peer assessment strategy has been based on the following principles.

- Students should read the course text and reflect on its content, so the peer assessment exercises asked a series of questions directly related to the text.
- Each student who answers the question becomes a reviewer of another student. The student reviewer should as their peer amend and correct the answers to the text-related questions instead of trying to figure out *how* the answer is incorrect.
- Students should be encouraged to participate. The incentive was that their answers to text questions would be available for them to use at the written exam. No other textual aids would be allowed at the exam. The collection of answers was kept in the form of a *portfolio* for each student. The exam questions were the same for all students, regardless of their participation in peer assessment or lack thereof.
- Students should use LaTeX for their answers to text-related questions, as this is a system that they are familiar with is well-suited for technical writing. A further advantage has been that the web-based implementation could easily handle the text files used by LaTeX. To this end, the students would be supplied with a standard preamble that they must use.

Some examples of text-related questions (translated from Danish) can be found in Figure 1.

(From the session about regular expressions) Let R be a regular expression. Is it always the case that the regular expression $R \cup \varepsilon$ denotes the same language as R? If the answer is yes, then explain why. If the answer is no, then provide a counterexample.

(From the session about scope rules described using structural operational semantics) Here is a big-step transition rule. Explain which kind of scope rules are captured by this particular transition rule.

$$(\textsc{call-1})\quad \frac{env_V, env'_P \vdash \langle S, sto\rangle \to sto'}{env_V, env_P \vdash \langle \texttt{call}\ p, sto\rangle \to sto'}$$

$$\text{where } env_P\, p = (S, env'_P)$$

Figure 1: Examples of text-related questions used (translated from Danish).

4.2 The Workflow of an Answering Session

Each of the 15 answering sessions of the course had the following workflow:

1. After the lecture, the teacher would publish a LaTeX file Doc_0 with questions about the associated part of the text to the course webpage.
2. No later than by the morning of the session, at deadline d_1, the student would then answer the questions found in Doc_0 and hand the answers in the form of Doc_1, an amended version of Doc_0.
3. Just after the deadline d_1 the Doc_1 files would be distributed among the participating students for commeting and reviewing. Each reviewer compiles Doc_1, read it and amend/modify the answers. This would then lead to a new version of the LaTeX file, Doc_2.
4. The students would finally get back Doc_2 and be able to obtain it and either use it as their own final version or revert to the one that they originally wrote. The final outcome of the session is then a file Doc_3 which is added to the portfolio of the student.

This workflow is summarized in Figure 2.

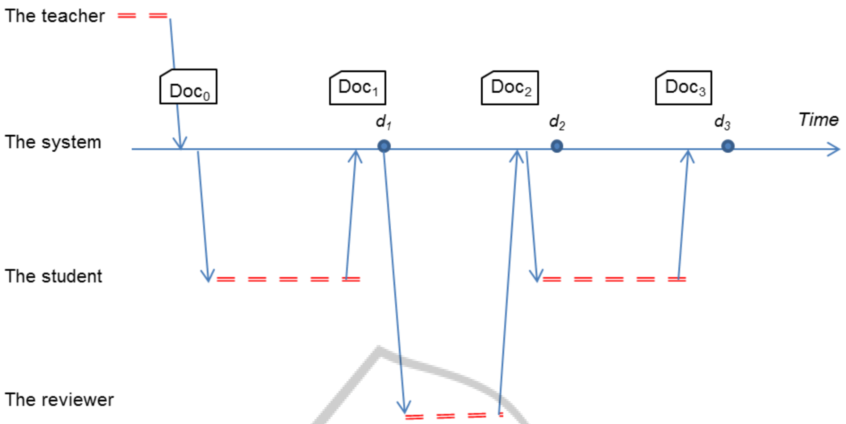

Figure 2: The workflow supported by the system, showing the documents that flow between the teacher, author, and reviewer together with the deadlines that apply.

4.3 The Implementation

The role of the system was to support the workflow described above, and make it possible to manage the portfolio of each student in an easy manner.

The system allows each student to log into a *status page* for each lecture in the course (with use of a valid username and password). Until the given deadline d_1 the status page allows upload of the LaTeX file Doc$_1$. After the deadline the system appoints a reviewer for each student that has submitted a Doc$_1$ in the current round of questions. The allocation of reviewers is done by computing a permutation of the students that have submitted a Doc$_1$ for this session. If possible, the permutation is formed in such a way that the reviewer assigned to a given student is in another project group than that of the original author.

With the reviewers appointed, the status page of the student S is augmented with information about the reviewer of S, $R(S)$, and the student that will be reviewed by S. In addition, $R(S)$ gets access to the material he or she is supposed to review via his or her status page together with a new deadline d_2. When a revised Doc$_2$ is submitted by $R(S)$, the original author S receives an email from the system, and the review is made available at the status page for S. If or when a review is received, S can submit a final version Doc$_3$, which allows S to elaborate the original contribution and/or the changes made by the reviewer $R(S)$.

The contribution to the portfolio of student S in the current round becomes Doc$_3$ if it exists, else Doc$_2$ if it exists, else Doc$_1$ if it exists, else the empty document.

The system provides the course teacher with an *overview page*, in terms of a table that shows the status of all students for every answering session so far. The teacher can activate the appointment of reviewers (construction of the permutation mentioned above) from the overview page. In addition, the teacher has access to the status page of each student via the overview page, and the individual LaTeX documents submitted by authors and reviewers.

At any point in time, it is possible for a student to construct an *aggregated portfolio* of answers to all questions. Behind the scenes the web system processes a LaTeX file, which is aggregated by the relevant pieces submitted by the author and the reviewers of the author. If there are problems with the processing (due to errors in the LaTeX document), the student is supposed to deal with these via the LaTeX log file, which also is made available.

The administrative system setup was mainly done programmatically, via Lisp text files. The system needs information about all participating students (full name, email address, encrypted password, and group number id). In addition, the system needs information about deadlines for each lecture, as illustrated in Figure 2). The system relies on a list of all d_1 deadlines (relative to Figure 2). The other deadlines are calculated by use of fixed offsets. In a future version of the system it will be attractive to have a web-interface to both student administration and the temporal setup of deadlines.

The current system is implemented (by the second author) in R5RS Scheme (Kelsey et al., 1998) with use of LAML (Nørmark, 2005) for CGI programming purposes. Administrative data are stored in files - one file per student per answering session.

5 EXPERIENCES

In this section we assess the experiences that we have had using the system in our teaching.

No answer at all	136
Only first answer	413
First answer and review	213
First and second answer, no review	67
First answer, review, and second answer	164
Only second answer	27
Total	**1020**

Figure 3: Distribution of the answering sessions.

5.1 A Quantitative Analysis

We have performed a quantitative analysis of data by using the information in the implemented web system juxtaposed with grades obtained by the students that attended the written exam. 68 students participated in 15 rounds of answering questions, leading to a total of 1020 *answering sessions* each of which follows the workflow shown in Figure 2. Figure 3 shows a classification of the 1020 potential answering sessions.

Of the 68 students, 26 were Computer Science students, and as such they participated in the written exam, where they got a grade from the Danish 7-step scale.[1] Figure 4 lists the number of answers (ranging from 0 to 15) and the corresponding number of students who submitted that particular number of answers. In addition, we show the list of grades obtained by the computer science students who were required take the exam. Thus, for instance, 15 students submitted 14 answers, and among these, 6 participated in the exam where they obtained the grades -3, -3, 4, 4, 7, and 10, respectively. One immediately notices that the majority of the students, who attended the written exam, submitted answers in almost all of the 15 rounds. This is less pronounced for software students, who did not participate in the exam.

The activity of each student can be measured in terms of the number of answers provided in total (0 .. 15), or more fine-grained, using the total number of answers (0, 1, or 2) and reviews (0 or 1) submitted for each answering session.

On the other hand, Figure 3 reveals that as many as 57 % of the answers provided were not followed by a review. This indicates that the aspect of peer assessment was not seen by the students as a necessary component of the teaching activity.

5.2 A Qualitative Analysis

By using the system and interacting with the students, we have made some observations of a more qualitative nature.

Number of answers	**Number of students**	**List of grades obtained (Computer Science students only)**
0	1	00
1	0	*none*
2	1	*none*
3	0	*none*
4	0	*none*
5	1	*none*
6	0	*none*
7	3	00
8	1	*none*
9	2	*none*
10	2	4
11	3	7
12	4	*none*
13	3	*none*
14	15	-3 -3 4 4 7 10
15	32	00 00 02 4 4 4 7 7 7 7 7 10 10 10 10 12

Figure 4: Exam results.

Unfortunately, it turned out that again, some students spent more effort trying to circumvent the system than in engaging in the learning process. As already noticed, some students uploaded answers but did not comment on the answers provided by others. This caused a considerable amount of frustration. Moreover, a couple of students discovered that the implementation accidentally allowed them to submit Doc$_3$ without submitting Doc$_1$. (This is the cause of the row 'Only second answer' in Figure 3). This meant that they could avoid the first deadline and also manage to avoid being involved in the commenting round.

Other evidence suggests that students were not always careful in producing their answers. Before the exam, all portfolios were generated and compiled by the system. In some cases, the students had systematically modified the standard preamble, making it incompatible with the script for generating portfolios. In several other cases, the LaTeX system was unable to generate a portfolio due to simple LaTeX errors that suggest that the author had never bothered to compile

[1] The Danish 7-step grading scale is compatible with the ECTS scale: 12 = A, 10 = B, 7 = C, 4 = D, 02 = E, 00 = Fx, -3 = F. The grade 02 is the lowest passing grade.

the LaTeX file before submitting it.

It turned out that strict adherence to all deadlines was difficult to accomplish. Illness and other unforeseen circumstances led to introduction of a period of time, where students could submit missing contributions. It turned out that 8% of all contributions were received during that period. 92% of the contributions were received on schedule, relative to the deadlines of the answering sessions.

In feedback from some of the students it is pointed out that the many deadlines that had to be met, were a considerable burden. The work hereby enforced on the students took time and efforts away from other courses, and from the group project. In addition, it was pointed out that the students wanted to be able to revise their contributions, even after the last deadline of an answering sessions. All taken together, it is the impression of the students that the approach taken was demanding.

6 CONCLUSIONS AND IDEAS FOR FURTHER WORK

We have described a web-based system for peer assessment. Our analysis of the exam results indicate that there is a statistically significant correlation between activity in the course and the result obtained at the examination.

The connection between coursework and the exam that follows is a powerful incentive for the students to answer the questions and to aggregate a portfolio that helps them during the exam. On the other hand, the fact that more than half of the answers submitted were not reviewed indicate that an important aspect of the approach, namely that of peer assessment, did not work out as planned. In order to improve the motivation for participating in the peer review work, we consider to change the rules in the next round of use. Our suggestion is is that a student can only have a set of answers to text questions added to the portfolio if the student has also submitted a review within the same session.

Developing the administrative aspect of our system is a topic of further work. Our system was simple in that respect; unforeseen events at student level were dealt with on an ad hoc basis. More importantly, the submitted files were never checked for LaTeX errors. In a future version of the system, the day-to-day administrative tasks and LaTeX issues should be supported directly by the web-based system. An approach to this is to integrate our ideas into an existing web-based system for handling teaching activities. As Aalborg University is currently in the process of switching its web-based course administration interface to Moodle (Moodle,), the ideas of our system will be integrated into a plugin for Moodle.

REFERENCES

Anderson, J., Austin, K., Barnard, T., and Jagger, J. (1998). Do third-year mathematics undergraduates know what they are supposed to know? *International Journal of Mathematical Education in Science and Technology*, 29(3):401–420.

Biggs, J. and Tang, C. (2007). *Teaching for Quality Learning at University (Society for Research Into Highter Education)*. Open University Press, 3 edition.

de Raadt, M. Michael de Raadt's Peer Review Assignment Type. http://www.sci.usq.edu.au/staff/deraadt/peerReview.html.

Kelsey, R., Clinger, W., and Rees, J. (1998). Revised^5 report on the algorithmic language Scheme. *Higher-Order and Symbolic Computation*, 11(1):7–105.

Lauvås, P. and Jakobsen, A. (2002). *Exit eksamen – eller?* Cappelen. In Norwegian.

Maugesten, M. and Lauvås, P. (2004). Bedre læring av matematikk ved enkle midler? Rapport fra et utviklingsprosjekt. Technical Report 6, Høgskolen i Østfold. (in Norwegian).

Moodle. Moodle. http://moodle.org.

Nørmark, K. (2005). Web programming in Scheme with LAML. *J. of Funct. Prog.*, 15(1):53–65.

Sitthiworachart, J. and Joy, M. (2004). Effective peer assessment for learning computer programming. In *9th Annual Conference on the Innovation and Technology in Computer Science Education (ITiCSE 2004)*, pages 122–126.

Topping, K. J. (2009). Peer assessment. *Theory Into Practice*, 48(1):20–27.

Ward, A., Sitthiworachart, J., and Joy, M. (2004). Aspects of web-based peer assessment systems for teaching and learning computer programming. In *3rd IASTED International Conference on Web-based Education (WBE 2004)*, pages 292–297.