Electronic Peer Review and Peer Grading in Computer-Science Courses

Edward F. Gehringer
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7911
efg@ncsu.edu

Abstract

We have implemented a peer-grading system for review of student assignments over the World-Wide Web and used it in approximately eight computer-science courses. Students prepare their assignments and submit them to our Peer Grader (PG) system. Other students are then assigned to review and grade the assignments. The system allows authors and reviewers to communicate with authors being able to update their submissions. Unique features of our approach include the ability to submit arbitrary sets of Web pages for review, and mechanisms for encouraging careful review of submissions. We have used the system to produce high-quality compilations of student work. Our assignment cycle consists of six phases, from signing up for an assignment to Web publishing of the final result. Based upon our experience with PG, we offer suggestions for improving the system to make it more easily usable by students at all levels.

1 Peer Review in the Classroom

Peer review is a concept that has served the academic community well for several generations. Thus, it is not surprising that it has found its way into the classroom. Dozens of studies report on different aspects of peer review, peer assessment, and peer grading in an academic setting. A comprehensive survey can be found in Topp 98. Experiments with peer assessment of writing go back more than 25 years.[4] Peer review has been used in a wide variety of disciplines, among them accounting,[8] engineering,[7, 10] mathematics,[3] and mathematics education.[6]

SIGCSE 2001 2/01 Charlotte, NC, USA

However, *electronic* peer review experiments have been much rarer. Although the Daedalus Integrated Writing Environment[1] is widely used for peer assessment of student writing, only a few computer-mediated peer-review experiments have taken place in other fields. An early project in computer-science and nursing education was MUCH (Many Using and Creating Hypermedia).[9, 11] The earliest reported software program to support peer evaluation was evidently created at the University of Portsmouth.[12] The software provided organizational and record-keeping functions, randomly allocating students to peer assessors, allowing peer assessors and instructors to enter grades, integrating peer- and staff-assessed grades, and generating feedback for students. One of the early Web-based peer-review experiments was described by Downing and Brown.[2] Their psychology students collaborated to create hypertexts which were published in draft on the World Wide Web and peer reviewed via e-mail. Our project is apparently the first to use the Web for both submission and review of student work.

2 Peer Review on the Web

There is much to recommend a Web-based approach to peer review. Unlike software that is written for a specific academic field (e.g., English composition), a Web-based application can accept submissions in practically any format, including diagrams, still pictures, interactive demonstrations, music, or video clips. Of course, the student has to understand how to produce such a submission, but for each field, that expertise tends to "come with the territory."

Secondly, the Web is a familiar interface. Most students use the Web in their day-to-day studies, so they can pick up a Web-based application for peer review with minimal effort. In addition, many if not most students are already familiar with tools for producing Web pages; for example, most wordprocessors can now save files in HTML format.

Thirdly, Web creation skills are of increasing importance in business as well as academia. In producing work for Web-based peer review, students not only learn about the subject

of their submission, but also gain valuable experience with software they will use in their later studies and on the job.

Fourthly, a Web interface enables the peer-review program to be used in distance education, which is an important and rapidly growing segment of the education market. On-campus students can review distance-education students, and vice versa, bringing the two groups closer together in their educational experience. With Web-based submission, there is no extra overhead for the instructor or TAs in handling distance-education students.

Finally, Web-based peer review facilitates the production of Web-based resources. The best peer-reviewed work can be turned into materials to help future classes learn. For example, students can write research papers on various topics, with several students writing on the same topic. The best paper on each topic can then be presented to the next semester's students as background reading on that topic. The writers can be asked to include liberal doses hyperlinks in their papers, so that later students can read not only their work, but also the analyses of experts. As another assignment, students can be asked to compose problems on the class material; the best of these can then be assigned to later classes as homework or test questions.

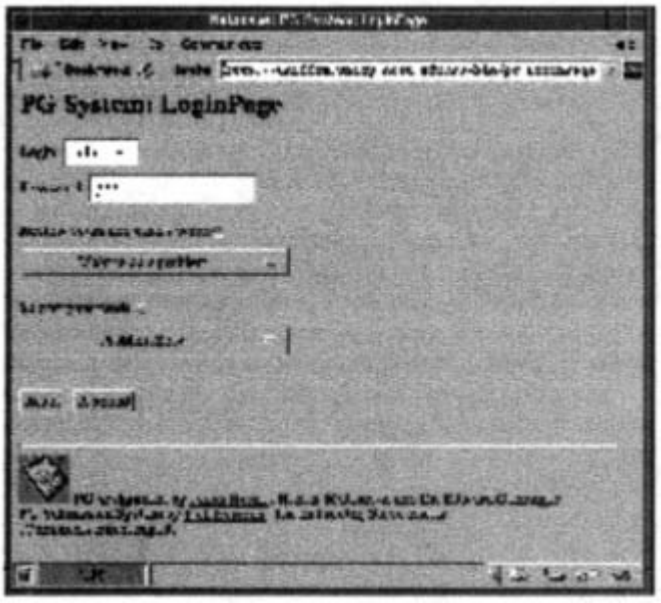

Figure 1: PG's login page

3 The PG System

PG is a portable Web-based application for peer review and grading written in Java. Originally it was implemented as a standalone Java application running under CGI, but it has recently been revised to be a servlet-based application. Students submit their work over the Web. Reviewers can be assigned pseudo-randomly by PG, or by the instructor, using a spreadsheet. The number of reviewers is arbitrary, but usually three or four students are assigned to review each submission. Reviewers and authors communicate double-blindly via a shared Web page. At the end of the review process, the reviewer assigns a grade to each author whose work (s)he has reviewed. A student's grade is the average of the grades given by the reviewers, plus an incentive described below to encourage careful reviews.

A student entering the PG system (Figure 1) has a choice of whether to submit a new page or review pages submitted by others. If more than one Web page is to be submitted, they may be submitted sequentially, each with a different filename, or submitted in a single Zip file, which PG will unpack into its components. Entire directory hierarchies may be submitted in this manner. Since the files themselves are copied, all work to be reviewed will have a URL beginning with the pathname of the PG system, not the submitter. This ensures that the reviewers will not be able to guess their authors' identities by dissecting the URL. The ability to submit directory hierarchies allows large projects to be submitted as easily as small ones.

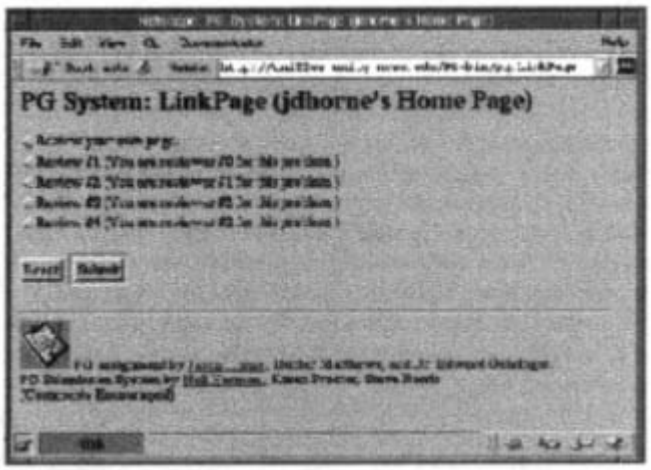

Figure 2: Page with links to submissions to be reviewed

Reviewers communicate with their authors via a shared Web page. There is one such page for each author (Figure 2); the author can view the reviewers' comments and vice versa. The instructor can configure the system either to allow (Figure 3) or not to allow reviewers to see the other reviewers' comments and assigned grades. There are reasons in support of both strategies. Allowing reviewers to see each other's feedback provokes better dialogue over the quality of a submission, but the first reviewer's comments may unfairly influence subsequent reviewers' assessments.

4 The Submit-Review-Publish Cycle

Our experience with PG has led us to a four- to six-phase cycle, capable of producing high-quality peer-reviewed work suitable for Web publication.

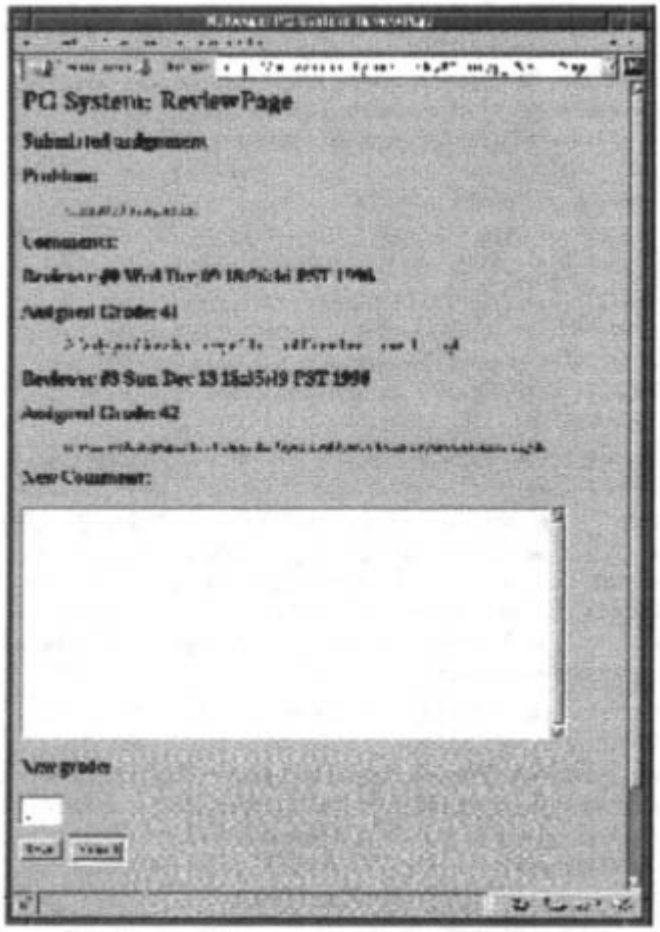

Figure 3: Review page

1. The *signup* phase (optional): If not all students are to do the same assignment, the students are given a list of potential topics (relating to research, or to a particular lecture, etc.) and sign up for one of them. To assure that all topics are chosen, only a limited number of students is allowed to sign up for any particular topic.
2. The *submit* phase. Students prepare their work and submit it to PG.
3. The *initial feedback* phase. Students are given a certain period of time—usually 3 to 7 days—to make initial comments on all the work. This phase was instituted after students complained that their reviewers often did not comment on their work until it was too late to revise it. Reviewers may assign a grade during this period, but they are not required to do so.
4. The *grading* phase. During the next period—again usually 3 to 7 days—students can revise their work in response to reviewers' comments, and reviewers can comment on the revisions. At the end of this give-and-take, reviewers are required to assign a grade. This grade is one component of the author's final grade for the assignment.
5. The *review of review* phase. After the review period is over, each student is presented with a set of reviews to assess. The students grade each *review* based on whether it was a careful and helpful review of the submission. The grades the students receive on their *reviewing* is then factored into their grade for the assignment (usually 25% of their grade is based on their reviewing). This phase was instituted after it was discovered that many students were doing cursory reviews. Since this approach was adopted in Summer 1999, the volume of communication between students and their reviewers has increased by 15%–35% (n = 733, with 459 before Summer '99), though direct comparisons are difficult because the courses and assignments before the change were different from those after the change. Qualitatively, the students also seem to be making more thoughtful comments than before.
6. The *Web publishing* phase (optional). PG creates a Web page with links to the best student assignment in each category. As described below, this can serve as a useful study tool for future generations of students.

5 How Peer Review Has Been Used

There are opportunities to use peer review in almost any computer-science course, from first-semester programming up to the graduate research level. The author has used it in courses ranging from second-semester programming to graduate reading courses.

Peer review can be used for *researching lecture material*. For example, the students can be assigned to find links to Web pages related to a particular lecture, with each lecture chosen by a different set of students. The best submissions can then be combined into an index that all students can use for studying. Second-semester programming students found many useful links to instructional materials on C++.

Taking this one step further, if the instructor has on-line lecture notes, the students can be assigned to *annotate* these notes with Web links in appropriate places. The compiled result is a valuable resource for in-depth study of the material. The author has done this in computer architecture, operating systems, and object-oriented systems courses.

Peer review can be used for *researching beyond the lecture material*. In the author's operating-systems course, each student selected a research topic from a set that included topics like "Scheduling in Windows NT," "Deadlock handling in Unix or a particular flavor of Unix," and "Virtual memory in Linux." Students in the author's computer ethics course have chosen among many topics for research, including "Privacy on the Web," "Antitrust: The Microsoft case," "MP3s," and "Access for the disabled." In the latter case, we used the best of these pages on our Ethics in Computer Website, http://www2.ncsu.edu/eos/info/computer_ethics.

Peer review can be used for *reviews of papers from the literature*. An excellent way for advanced students to gain in-depth knowledge of a subject is for them to read and comment on research papers. In most courses, the author has graded these reviews himself, but recently he has started using peer review for this assignment. The students give more feedback than the author or TAs are capable of after reading 80 papers. Students can be assigned to review students who have read different sets of papers, giving them valuable (though second-hand) exposure to additional research areas.

Though it might not be obvious at first glance, there are many ways to use peer review in *programming courses*. Students can be assigned to do design reviews of each other students' semester projects. This helps to familiarize the students with object-oriented design and analysis principles, and also helps the students improve the design of their own projects. When the author teaches design patterns, he has each student identify a design pattern in code that (s)he has written and present it to other students. The other students grade it based on how well it illustrates the design pattern in question.

In almost any course, students will learn by *making up a problem* on the material covered in the course. These are very time consuming for the instructor to grade, but students will learn by peer-reviewing other students' problems. As a side benefit for the instructor, about one-quarter of the student submissions are good enough to be used on future problem sets or tests.

Finally, the author has used peer review for *weekly reviews* of independent-study students. Each summer he has students update and enhance the Ethics in Computing Website. Weekly peer reviews are an excellent way to insure that students do their work on time, throughout the semester, instead of trying to finish everything in the last week or two.

6 The Lessons of Experience

Our goal is to make PG a tool for Web-based education in many fields across the curriculum, and in high school as well as in higher education. To do this, we need to design a "bulletproof" interface that does not take a techie to use it successfully. Several semesters of experience have identified these pitfalls.

Students don't submit `index.html` files, even if the program warns them they must. Since PG accepts arbitrary sets of Web pages, it must be told where to start. PG uses the standard Web convention of beginning with the `index.html` page in the top-level directory. Even though it warns students at least twice to submit an `index.html` file, many don't. Or they submit a file called `index.html` that is not linked to the rest of their submission. It would probably be better to have PG begin with a file of any name if there is only one file in the top-level directory.

Students use absolute pathnames to hyperlinked (e.g., image) files. The link may point to the hard drive of the computer they submitted the file from. Such links, of course, won't work when their submission is reviewed. If they submit from the campus computer network, the link may work, but the pathname can give away the author's identity. However, we can't ban absolute pathnames entirely without giving up the ability to do lecture annotations and research papers with hyperlinked references to outside work.

Students sometimes put their names on their submissions. It should be possible to check for this and warn the student, though it requires PG to know students' names, not just their user-IDs.

For assignments requiring advance signups, students submit without signing up, or they submit something other than what they signed up for. On our agenda are modifications to prevent a student from submitting without signing up, and to put the title of the signed-up-for assignment on the page that links to the submission, so reviewers will instantly know if the student has submitted the wrong assignment.

Students select the wrong assignment from the dropbox listing assignments, and wonder why they cannot log in. Our solution to this is to generate static HTML for the entry page of each assignment, so that students can go directly to the login page for their PG assignment without having to select from a list. This is already in use for the signup sheets, and seems to solve the problem.

Some students prefer to do reviews by e-mail instead of over the Web. Since PG already e-mails reviews to authors as they are posted on the Web, this would be a fairly easy extension, although it has not been done yet.

It is often difficult to determine who exactly is in the class. This is a problem for Web-based education in general, not just for PG. Students may register late, or they may fail to read their campus e-mail *and* fail to tell the system which e-mail address to use for them, and consequently may not receive e-mail telling them about class assignments.

On the other end, it causes problems for peer review when a student drops the course after an assignment has been assigned but before reviews have been completed. This results in some students not having enough work to review and others not getting enough reviews of their work. The problem is exacerbated when students decide early to drop a course and then wait until right before the deadline to drop it. Thus, it is never possible to know with certainty which students can be expected to participate in an assignment. One solution is to dynamically map authors and reviewers; that is, to postpone assigning reviews until the point when a student logs in and asks to do a review. This is possible, but tricky. The fact that a student has *asked* to do a review doesn't guarantee that the review will get done; perhaps the reviewer is interrupted and never returns to the task. So, after some idle time, we should put

the review back into the pool of reviews to be done. But how much time is enough? Secondly, we must map reviewers to authors with care, so that we never get to a point where a student can only be assigned to review himself. This problem is more likely to crop up during the review-of-review period; students must not be assigned to review themselves, nor may they be assigned to review reviews of their own work.

Another difficulty is the student who does not submit on time, leaving his/her reviewers too little time to respond. This is not always the student's fault; in distance-education courses, especially, people do get sent on business trips, and Internet access is not always available, nor does everyone travel with a laptop. So, although the author's students have accessed PG from Colorado to Bahrain, some provision should be made for the student who is temporarily out of contact. This suggests that PG should incorporate the ability for students to negotiate review deadlines with their reviewers, or to do review mappings in groups based on when students have initially submitted. However, since other considerations often constrain the choice of reviewers,[5] it may be impossible to form dynamic groups for some assignments.

7 Future Development

PG has been developed mainly by students working on independent-study and Senior Design projects. But recently, the author has begun to assign some PG projects as semester projects in his graduate-level object-oriented systems class. The designs for these projects have been reviewed by other students in the class—using PG, of course. Good code has been produced in this way, but it has required too much time to integrate all of the projects into the PG system. This year, the author plans to use pair programming[13] for some of the projects. This will be an interesting experiment in its own right, but more to the point, it will reduce the substantially reduce the number of projects that need to be integrated relative to the amount of code produced, thus diminishing the task of integration.

8 Conclusion

We have developed software for peer review and peer grading over the Web. This software has been used in in eight courses, with good results. In three different courses, students were asked whether peer review was helpful to the learning process. The average response was 3.57 to 4.24, on a scale of 1 to 5, with 5 being very useful. (Further details can be found in Reference 5.) We have demonstrated that there are many different ways to use peer review in computer-science classes. We are readying PG for wider distribution in 2001, and are seeking collaborators in a variety of different fields at several different levels in the curriculum.

References

[1] The Deadalus Group, Daedalus Integrated Writing Environment, http://www.daedalus.com/info/overtext.html

[2] Downing, T. and Brown, I., "Learning by cooperative publishing on the World-Wide Web," Active Learning 7, 1997, pp. 14-16.

[3] Earl, S. E., "Staff and peer assessment: Measuring an individual's contribution to group performance," *Assessment and Evaluation in Higher Education* 11, 1986, pp. 60-69.

[4] Ford, B. W., *The effects of peer editing/grading on the grammar-usage and theme-composition ability of college freshmen.* Dissertation Abstracts International, 33, 6687.

[5] Gehringer, Edward F., "Strategies and mechanisms for electronic peer review," *Proc. Frontiers in Education 2000*, Kansas City, October 18–21, 2000 (to appear).

[6] Lopez-Real, Francis and Chan, Yin-Ping Rita, "Peer assessment of a group project in a primary mathematics education course," *Assessment and Evaluation in Higher Education* 24:1, March 1999, pp. 67-79.

[7] MacAlpine, J. M. K., "Improving and encouraging peer assessment of student presentations, *Assessment and Evaluation in Higher Education* 24:1, March 1999, pp. 15-25.

[8] Persons, Obeua S., "Factors influencing students' peer evaluations in cooperative learning," *Journal of Business for Education*, Mar.–Apr. 1998.

[9] Rada, R., Acquah, S., Baker, B., and Ramsey, P., "Collaborative learning and the MUCH System," *Computers and Education* 20, 1993, pp. 225-233.

[10] Rafiq, Y., & Fullerton, H., "Peer assessment of group projects in civil engineering," *Assessment and Evaluation in Higher Education* 21, 1996, pp. 69-81.

[11] Rushton, C., Ramsey, P., and Rada, R., "Peer assessment in a collaborative hypermedia environment: A case-study," *Journal of Computer-Based Instruction* 20, 1993, pp. 75-80.

[12] University of Portsmouth, "Transferable peer assessment," in National Council for Educational Technology [ed.], Using information technology for assessment, recording and reporting: Case study

[13] Williams, Laurie A. and Kessler, Robert R., "All I really need to know about pair programming I learned in kindergarten," *Communications of the ACM* 43:5, May 2000, pp. 108–114.