AC 2010-1154: MOTIVATING EFFECTIVE PEER REVIEW WITH EXTRA CREDIT AND LEADERBOARDS

Edward Gehringer, North Carolina State University

Ed Gehringer, efg@ncsu.edu, is Associate Professor of Computer Science and Computer Engineering at North Carolina State University. His main research area is collaborative learning technology. He received his Ph.D. degree from Purdue University, and taught at Carnegie Mellon University, and Monash University in Australia.

Abhishek Gummadi, North Carolina State University

Abhishek Gummadi, agummad@ncsu.edu, is a graduate student in the Department of Computer Science at North Carolina State University. He is working on a masters thesis on the role of game mechanics in motivating effective peer reviewing.

Reejesh Kadanjoth, North Carolina State University

Reejesh Kadanjoth, rkadanj@ncsu.edu, is a graduate student in the Department of Computer Science at North Carolina State University. He is working on a masters thesis on supporting the creation of large documents such as wiki textbooks in an online peer-review system.

Yvonne Marie Andrés, Global Schoolnet.org

Yvonne Marie Andrés, yvonne@gsn.org, is the President & CEO of the Global SchoolNet Foundation, which partners with schools, communities and businesses to provide online collaborative learning programs that prepare students for the workforce and help them to become responsible and literate global citizens.

Motivating Effective Peer Review with Extra Credit and Leaderboards

Abstract

Peer review is a pedagogically sound practice that has found its way into education in all fields, including engineering. Students are often skeptical of its value, and don't give it the attention it merits. We have implemented several facilities to address this need in our Expertiza peer-review system. Rubric-based review guides each student through the review process. Metareviewing is performed to assign scores to each reviewer's reviews. Scores for reviewing can be factored into the student's grade. Leaderboards allow students to see, with permission, who are the class leaders in categories such as submitted work and review quality. We are currently working on implementing an approach to awarding extra credit to reviewers for interacting with their authors more frequently than required.

1. Motivation

Online gaming plays an important role in the lives of many of our students. They compete with each other to earn points and achieve status. The rewards motivate them to spend hours working on task. Suppose we could entice our students to contribute to each other's learning experience by awarding points for providing formative feedback and answers to each other's questions. This, it turns out, can be accomplished by a simple extension to student peer review.

Expertiza [1, 2] is a Web-based system for peer-reviewing student work. After students submit their work by uploading a file or writing on a wiki, other students are assigned to review their work. Author and reviewer communicate in double-blind fashion using rubrics. The reviewer fills out a rubric that includes several questions about the author's work, as

00	response edit	
) × C × (1	() (http://expertiza.ncsu.edu/response/edit/11080 () * *)	Google
t Visited - Getting Starter	t Latest Headlines බ	
response edit		
		0000001111100010
Expertiza		
Reusable learning objects	through peer review	CAPERIIZA
ome Assignments Co	ourse Evaluation Profile Contact Us	
Home		User: arwaugam
Edit Dovio	w for Brogram 2 style	Payart
	with Flogram 2 Style	
Submitted Files hid	e files	
Name S	ize Type Date Modified	
** Program_2.zip 88	381 file Thu Nov 12 00:11:47 -0500 2009	
		_
Formatting. Is code pro	operly and consistently indented? Are braces used and located consistently? Is whitespace	
used for readability? A	show advice	
Commenting. Are com	ments used to explain what each class does? Do all nonaccessor methods have comments	
giving the purpose, na code within methods?	me, and parameter of the method? Are comments used to explain non-obvious sections o	r
	1 show advice	
Naming. Are names m	eaningful, and suggestive of the purpose for which the name is used?	
Naming. Are names m	eaningful, and suggestive of the purpose for which the name is used?	

Figure 1. Filling out a review form

shown in Figure 1. Each rubric question can be given a numeric score, and there is space for a prose comment on each question. The system computes a composite score based on the scores for the individual rubric questions.

While filling out a review rubric, the reviewer can be guided by rubric *advice*, which is a description of the characteristics of work meriting a particular score for a particular question. Clicking on "Show advice" will display the advice for the question. The reviewer can then click on the advice for a particular score, and that score will appear in the dropbox. Figure 2 shows how this works. We have begun to involve students in the process of creating rubrics, by having the class discuss first the rubric questions and then the rubric advice [3].

		response edit	
) b (C) (X) (🔒 (http://es	xpertiza.ncsu.edu/response/edit/11080	☆▼_) - (<mark>*] • (</mark> Google
t Visited ~ Getting	started Latest Headlines ?	9	
response			
Edit Rev	iew for Pr	ogram 2 style	Revert
Submitted Files	hide files		
Name	Size Type	Date Modified	
Formatting. Is co used for readabili Commenting. Are giving the purpos	le properly and consist ty? Are too-long-lines g comments used to exp e, name, and paramete	ently indented? Are braces used and located consi generally avoided? show adv blain what each class does? Do all nonaccessor me er of the method? Are comments used to explain n	stently? Is whitespace ice thods have comments on-obvious sections of
Jode widnin medi	JUS ?	4 i hide advi	ce
LIICK advice to	adjust score		
1: Almost no co	mments, and where the	ey exist, they do not help in describing the workin	g of the code.
1: Almost no co 2: Some metho not explained by	mments, and where the ds are commented, but r comments.	ey exist, they do not help in describing the workin t generally not in Javadoc format. Much of the wor	g of the code. king of the code is
1: Almost no co 2: Some metho not explained by 3: Most method body of the cod playlist).	mments, and where the ds are commented, but comments. s are commented with There are few comm	ey exist, they do not help in describing the workin t generally not in Javadoc format. Much of the wor explanations of how they work. Long methods have ents that simply repeat what the code does (playli	g of the code. ding of the code is e comments in the st.sort(); // Sort the
1: Almost no co 2: Some metho not explained b 3: Most method body of the cod playlist). 4: Most method cases (null para comments that	with the second	ey exist, they do not help in describing the workin t generally not in Javadoc format. Much of the wor explanations of how they work. Long methods have ents that simply repeat what the code does (playli explained in proper Javadoc exited. Behavior is dess Most sufficiently complicated code is commented. code does (playlists.sort()) // Sort the playlist).	g of the code. king of the code is te comments in the st.sort(); // Sort the ribed in many special There are no
1: Almost no co 2: Some metho not explained b 3: Most method body of the cod playlist). 4: Most method cases (null para comments that 5: All methods i Javado cstyle. T complicated coc (playlist.sort();	mments, and where th ds are commented, but / comments. s are commented with e. There are few comm s are commented and meters? empty lists?). simply repeat what the with the possible excep- here are comments in 1 e is commented. There (/ Sort the paylist).	ey exist, they do not help in describing the workin t generally not in Javadoc format. Much of the wor explanations of how they work. Long methods have tents that simply repeat what the code does (playli explained in proper Javadoc style. Behavior is dess Most sufficiently complicated code is commented. code does (playlists.sort(); // Sort the playlist). ption of getters and setters) are explained and cor the bodies of methods longer than 10 lines or so, a are no comments that simply repeat what the co	g of the code. ding of the code is e comments in the st.sort(); // Sort the ribed in many special There are no amented in proper and any sufficiently de does
1: Almost no co 2: Some metho not explained bi 3: Most method body of the cod playlist). 4: Most method cases (null para comments that 5: All methods is 5: All methods is complicated coc (playlist.sort();	mments, and where th ds are commented, but y comments. s are commented with s. There are few comm s are commented and q meters? empty lists?), simply repeat what the with the possible excep here are comments in I e is commented. There // Sort the playlist).	ey exist, they do not help in describing the workin t generally not in Javadoc format. Much of the wor explanations of how they work. Long methods have ents that simply repeat what the code does (playli explained in proper Javadoc style. Behavior is desc Most sufficienty complicated code is commented. code does (playlist.sort(), '/ Sort the playlist). ption of getters and setters) are explained and cor the bodies of methods longer than 10 lines or so, a re no comments that simply repeat what the co	g of the code. king of the code is te comments in the st.sor(); // Sort the ribed in many special There are no mmented in proper and any sufficiently le does

Figure 2. Using rubric advice

Similarly, authors give feedback to their reviewers, using a rubric that asks about the helpfulness, accuracy, and respectfulness of the review; and the scores from this feedback can be included in the reviewers' grades. The form appears when the author views a review (Figure 3).

If we keep track of these scores for a whole class, we can display them—with the student's permission—to others via a *leaderboard*. A leaderboard is simply a list of the leaders in a particular category, such as the students receiving the highest scores on reviews of their work, or the students whose feedback is rated most highly by the authors they have reviewed (Figure 4). Students will only appear on the leaderboard if they give permission (by editing their profile) for their scores to appear.

In Expertiza, grades can be based on a number of factors, including reviews by other students, metareviews, and feedback from the authors who are being assessed. The weightings are specified when an assignment is set up (Figure 5). The fact that their

\varTheta \varTheta 🔿 response new	2
(
Most Visited ~ Getting Started Latest Headlines A	
response new +	i,
Question 2: Commenting. Are comments used to explain what each class does? Do all nonaccessor methods have comments giving the purpose, name, and parameter of the method? Are comments used to explain non-obvious sections of code within methods?	
(⁵ out of 5): Lots of comments, but not too many in number. They explain the classes and anything that may be unclear.	
Question 3: Naming. Are names meaningful, and suggestive of the purpose for which the name is used?	
(5 out of 5): Very meaningful names, they are clear about what they represent.	
Question 4: Visibility. Is visibility of members (variables, methods, classes) limited to the parts of the program that need to access them (or test them)?	
(<mark>5</mark> out of 5): Yes.	
Question 5: Structure. Are classes cohesive, with low coupling between them? Or, is it necessary for one class to repeatedly call accessor methods of other classes, when that could have been avoided by placing functionality in another class?	
(5 out of 5): Very cohesive, and it is not necessary for one class to repeatedly call accessor methods of other classes when it could have been avoided.	
Question 6: Testing. Are there tests for all classes? Do the tests test the obvious boundary cases?	
(<mark>5</mark> out of 5): Yes.	
Additional Comment:	
This reviewer appeared to understand my work.	
This reviewer's comments helped me improve my work.	
The tone of this review was respectful	
Additional Comments	Ļ
	Ŧ

Figure 3. Giving feedback to reviewer

reviewing counts in their grade gives the students an incentive to do careful reviews. If there are no serious disagreements between reviewers, or between reviewer and author, we normally accept student-assigned scores as the grade for the assignment. In case of controversy, however, the instructor or TA always has the ability to override studentassigned scores.

Expertiza Reusable learning objects through peer review							Expertiza		
Home Assig Home	nments Profile Conta	act Us							User: student1
Wiki Assignment 3 Leaders for this assignment							Reveil		
Reviews of their work			Author feedback			Melareviews			
Rank 1 2 3	Name UNC TLT Participant1 UNC TLT Participant8 UNC TLT Participant3	Average 93.34 90.0 86.67	Rank 1 7 3	Name UNC TLT Participant5 UNC TLT Participant1 UNC TLT Participant9	Average 100.0 90.0 80.0	Rank 1 2 3	Name UNC TLT Participant3 UNC TLT Participant10 UNC TLT Participant1	Average 95.0 90.0 90.0	
Submit	or review wor	k							
Your Your Other View Chang	work (Submit and view s' work (Give feedback scores(View feedback or ge your handle (Provide	your work) to others on n your work) a different h	their work andle for t) his assignment)					

Figure 4. A leaderboard

000		assign	iment new				
▼ C × (http://expertiza.ncsu.ed	u/assi	gnment/new?priv	ate=0	<u>े</u> र र 🕻	Google	٩
Most Visited - Getting Starte	d Latest Headlines ລ						
assignment nev	v (+)						1
					00000	111000111110001	
Expertiza					000000	111000111110001 11110001	7
Reusable learning objects	through peer review					EXPERTIZ	A
Home Manage Sur	vev Deployments Assignments (ourse	Evaluation Pro	ofile Contact Us			
Manage						User: efg	
New Assig	Inment					Logout	
		_					
Assignment name:	Program 2 style		Wiki Type:	No	\$		
Submission directory	csc216/f09/p2	_	Team assignm	nent: Yes	•		- 1
URL of assignment description:			participants p	er team:			- 1
	Selected questionnaire		Weight	Notification perc	ent		
Review:	Java program style	•	100 %	60 %			
Metareview:	Review of review	:	0 %	15 %			
Feedback:	Standard author feedback	•	0 %	10 %			
Teammate Review:	Standard teammate review	:	0 %	15 %			
Reviews should be vis	ible to all other reviewers 📃						
Users must sign up be	efore submitting						
Jone							

Figure 5. How the instructor can count reviewing toward a student's grade

2. Encouraging more frequent interaction

Author and reviewer are not limited to a single interaction. After receiving feedback from the reviewer, authors can revise their work, and reviewers can review it again. We are currently extending our system to track and give credit for these extra reviews. Providing formative feedback through extra reviews enhances learning, and those who give the most frequent reviews can be recognized with their own leaderboard category. To prevent credit for vacuous reviews, points are only awarded for reviews that are ranked above a certain threshold for helpfulness by the author.

Thus far, we have been talking about earning credit for assigned work—reviews, metareviews, teamwork, and the like. We are also interested in stimulating other kinds of interaction, such as asking and answering questions on a message board. We can allow readers to rate the questions and answers on a numeric scale, and award points for them, depending on how likely they are rated. However, rewarding unassigned work can lead to collusion. A group of students can get together and decide that they will rate each other's questions and answers highly, regardless of merit. Preventing collusion is one goal of a reputation system.

3. Reputation systems

A reputation system is a way of measuring the reliability of ratings. Scores assigned by reviewers and metareviewers can be factored into a student's reputation. Several algorithms [4, 5, 6] have also been published for determining reviewer reliability, based only on the scores assigned by reviewers. These algorithms consider (i) consistency of scores assigned by this reviewer with scores assigned by others to the same work, and (ii) spread, how much the highest score the reviewer assigned differs from the lowest score (s)he assigned. Some algorithms also consider (iii) leniency, the tendency of a reviewer to give scores that are higher than other reviewers. Research [6] demonstrates that these algorithms provide effective quality control for reviews.

A reviewer's reputation can be used in two ways: as part of a reviewer's grade for an assignment, or to weight scores assigned by this reviewer in calculating other students' grades. A score assigned by a reviewer with a good reputation should count for more than a score assigned by a poor reviewer when computing the grade of the student being reviewed. Finally, students with the best reputations can be recognized on the leaderboard, just as they can be recognized for good reviewing or for performing a lot of extra reviews.

A student's reputation should be based both on assigned work and unassigned work. It is harder to compute reputation for unassigned work, because the strategy must deal with collusion. One simple strategy is to base reputation primarily on assigned work, and allow unassigned work to affect it only slightly, so that collusion is not worth the trouble. Another approach, in use by Slashdot, is to base reputation on the number of reviewers, not reviews, that rate a student's work as excellent. Then regardless how many "5"s a reviewer gives an author, it is no more valuable than giving a single "5". A third way is to reward a reviewer for being the first to give a particular post the same rating it later receives from other users; this is a takeoff on the SPEAR algorithm [7] used to combat spam on social Web sites. It would have the effect of damaging the reputation of colluders who rate a question or answer more highly than others rate it.

Students can also be awarded points for answering student questions on a message board. The questioner can rate the helpfulness of the response, and so can other students. Helpful answers to questions is another leaderboard category. To prevent collusion between students who agree to rate each other's answers highly, points are awarded for the *number* of students who highly rate a particular student's message-board answers.

4. Administration

Setting up an assignment entails several steps.

- 1. Creation of rubrics for review, feedback from authors, teammate review, and metareview.
- 2. Creation of the assignment, by specifying the rubrics, deadlines, and type of submission to be accepted.
- 3. Adding the participants to the assignment.

- 4. Specifying the teams.
- 5. Specifying the review assignments (who reviews whom).

To make it easy to get started with the system, we have created several "public" rubrics of all four types. A new instructor can simply select from these rubrics, or copy and modify them, if desired.

Participants, teams, and review assignments can be imported from CSV files. The Expertiza project staff typically helps instructors set up their first assignment. After that, subsequent assignments can be created by copying and modifying an existing assignment.

Because the leaderboard system is new to Expertiza, we don't have empirical data on how it affects engagement and student learning. However, we do have a survey on the peer-review process that has been administered to students over several previous semesters. This will allow us collect data on engagement and learning before and after leaderboards were incorporated into the system.

5. Conclusion

Peer review holds important advantages for students, both as assessors and as assessees. As assessors, the spend time reviewing, summarizing, diagnosing misconceived knowledge, and considering deviations from the ideal. As assessees, they write for an audience where they have the burden of making themselves understood, rather than writing for an expert grader, who is expected to be able to decipher their intentions. A large number of studies attest to the value of academic peer review in a wide variety of disciplines [8]. Students, however, are very skeptical about the value of peer review to their learning [9]. A clear need is to motivate students to "buy into" the peer-review process. In Expertiza, we have implemented several mechanisms for motivating good reviews: (1) authors can pose questions to reviewers, to understand the review, and how the reviewer thinks their work can be improved; (2) a student's review can be *metareviewed* by the instructor or by another student, and a score assigned to it; (3) both feedback from the author and metareviews can be counted in determining a student's final grade on the assignment, and (4) students can be recognized on a leaderboard as being among the best reviewers in the class. Future plans include incorporating a way to give extra credit for more frequent reviews, and implementing a reputation system that will bring additional evidence to bear on evaluation of reviewer competence.

Bibliography

[1] Gehringer, E.F.; Ehresman, LM.; Conger, S. G.; Wagle, P. A., "Reusable learning objects through peer review: The Expertiza approach," *Innovate—Journal of Online Education* 3:6 (August/September 2007).

[2] Gehringer, E.F., "Expertiza: information management for collaborative learning," in *Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-Learning Support*, A. A. Juan Perez [ed.], IGI Global Press, 2009.

[3] Gehringer, E.F. "Teaching interactively with Google docs," *Proc. Am. Soc. for Engineering Education* 2010 Annual Conference

[4] Hamer, J., Ma, K. T., and Kwong, H. H. "A method of automatic grade calibration in peer assessment." In *Proceedings of the 7th Australasian Conference on Computing Education – Vol. 42* (Newcastle, New South Wales, Australia). A. Young and D. Tolhurst, Eds. ACM International Conference Proceeding Series, vol. 106. Australian Computer Society, Darlinghurst, Australia, 2005, pp. 67-72,

[5] Lauw, H. W., Lim, E.-P., and Wang, K. Summarizing review scores of "unequal" reviewers, in 2007 SIAM International Conference on Data Mining, Minneapolis, April 26–28, 2007, pp. 539–544.

[6] Cho, K. and Schunn, C. "Scaffolded writing and rewriting in the discipline: A web-based reciprocal peerreview system." *Computers & Education* 48:409–426. 2007.

[7] Noll, M. G., Au Yeung, C., Gibbins, N., Meinel, C., and Shadbolt, N. "Telling experts from spammers: expertise ranking in folksonomies." In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in information Retrieval* (Boston, MA, USA, July 19 - 23, 2009). SIGIR '09. ACM, New York, NY, 612-619.

[8] Topping, K. "Peer assessment between students in colleges and universities, "*Review of Educational Research* 68:3, Fall 1998, pp. 249-276.

[9] Keeney-Kennicut, W., Guernsel, A.B., and Simpson, N. "Overcoming student resistance to a teaching innovation," *J. for the Scholarship of Teaching and Learning*, January 2008, pp. 1–26, http://www.georgiasouthern.edu/iostl