Tag Based Feedback for Programming Courses

Stephen Cummins
Durham University
Department of Computer Science
South Road, Durham, England
Tel: +44191 33 41704
s.a.cummins@durham.ac.uk

Liz Burd
Durham University
Department of Computer Science
South Road, Durham, England
Tel: +44191 33 41719
liz.burd@durham.ac.uk

Andrew Hatch
Durham University
Department of Computer Science
South Road, Durham, England
Tel: +44191 33 41706
andrew.hatch@durham.ac.uk

ABSTRACT
This paper reports the findings of a preliminary investigation into whether feedback generated by annotating source code with tags is considered useful by undergraduate students. These types of annotations facilitate a new approach to presenting assessment feedback to students in the form of a Web 2.0 tagging environment. This paper highlights the benefits and limitations of this approach as well as details of student reaction and behavior. This investigation focuses on assessment and feedback for an undergraduate Software Engineering Group Project. The preliminary results collected encourage further investigation of this approach.

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer and Information Science Education – computer science education

General Terms
Design, Experimentation

Keywords
Assessment, Education, Feedback, Folksonomy, Programming

1. INTRODUCTION
Often feedback for programming assignments is given in the form of a separate evaluation sheet relating to the work that has been assessed. This is not the best approach to feedback delivery as this summary feedback is divorced from the actual code [1] that makes up the software project. This lack of context can cause student confusion and does not facilitate targeted improvements to learning programming.

This project aims to provide an alternative to traditional feedback by adopting the Web 2.0 concept of using tags to organize information, or in this case feedback for source code.

As part of this research two prototype systems were developed to support generation and dissemination of feedback. The first is a plug-in for the Eclipse Integrated Development Environment (IDE) enabling assessors to annotate student projects in a familiar IDE. The second is a web based feedback system that enables assessors and students to view feedback in the form of tags through a browser.

2. PROBLEM STATEMENT
The learning and teaching of computer programming is recognized as a difficult undertaking [2] throughout the disciplines of computer science education. One important axiom in educational literature is that good feedback is crucial for improving student understanding and achievement [3]. It is for these reasons that the project has focused on investigating the effects of introducing a new method of feedback generation and dissemination. This investigation aims to determine the suitability of this new approach to feedback and identify the benefits or limitations in terms of learning and teaching in computer programming courses.

This paper introduces the novel approach of using shorter annotations in the form of tags to comment on student programming work. Shorter annotations are often more general and easier to reuse. Reuse of feedback in this system is important for providing users a greater context for exploring their own feedback. These collections or clouds of feedback can represent an individual students' feedback or indeed the feedback of an entire cohort. The ability for students to share these feedback clouds along with the associated source code in a Web 2.0 style provides additional novelty.

Frequency analysis and co-occurrence of tags are techniques easily implemented when using tagging systems of classification. These may provide insights into patterns of feedback which occur in a cohort. This information could prove interesting to (i) course directors, to aid them in modifying their teaching and (ii) students, supporting direction in their learning.

3. REVIEW OF LITERATURE
Joy et al [4] discuss a semi-automated system for source code assessment called BOSS. This system allows assessors to use electronic criteria sheets whilst they mark source code to generate feedback for their students. A consequence of this approach is that the feedback is taken out of context of the students work, thus potentially reducing its understandability [1].

The annotation of source code as a form of feedback is not a new concept with Mason [1] and Sitthiworachart [5] illustrating two approaches. Mason et al introduces a system where by the

assessor can attach free form comments to preselected source code constructs or comment on the entire source file. This paper builds on this approach and provides a different perspective to source code annotation using Web 2.0 techniques.

4. INVESTIGATION METHOD

In order to evaluate the effectiveness of the feedback generated, an investigation conducted with second year undergraduates has been designed. Students working on a Software Engineering Group Project are participants in the investigation. During the final assessment each groups' source code is annotated with feedback using the following process.

Two assessors select the same two files from each project and separately annotate them using the eclipse plug-in. During the annotation process the assessors focus on code style, ease of understanding and appropriateness of design to generate tags that comment on and suggest improvements to the work.

In order to select suitable candidate files for assessment a method of sampling is used which involved choosing the two files that have the highest frequency of modifications as recorded from the projects configuration management software. This sampling process is used as a convenient method of choosing possibly interesting candidate files to assess. The frequency of changes is used to indicate the effort spent on a particular source file and of course may not always be representative.

Assessors are not given any specific training on how to generate tags for source code. Basic instruction in how to operate the IDE plug-in and the web based feedback manager is given to assessors before hand.

Students are given access to their feedback clouds with the associated source code as well as the opportunity to share and view other students' feedback and work. The sharing functionality is intended to allow students to gain greater benefit from the tags used not only in their own work but in other peoples. This also increases the amount of feedback that an individual can receive without significantly increasing the assessors' workload.

It should be noted that students' assessment scores for the module are not available in the feedback system; only the feedback tags and associated source code can be viewed and shared.

4.1 Data Collection

In order to investigate how students use the software and whether or not they choose to share their feedback and associated work; data logging functionality was added throughout the system. This records not only the frequency of access but also which aspects of the system were used and how often.

Students' feedback on the general approach will be gathered using electronic questionnaires. The questionnaires will focus mainly on the overall feedback approach instead of usability of the software.

4.1.1 Sample

All students who have taken part in the group project are given individual access to their groups' feedback.

The sample size consists of 67 undergraduates separated into 12 groups. The demographic of the sample is weighted in favor of males with 96% and only 4% females.

5. RESULTS

5.1 System Usage

58% (39/67) of the cohort logged in at least once to view their groups' programming feedback.

Table 1 - System Usage Data

Group	Shared Feedback	Number of System Logins	Group Size	Score %
A	No	8 from 6 people	6	78
B	Yes	8 from 5 people	6	64
C	Yes	7 from 4 people	6	84
D	Yes	6 from 3 people	5	88
E	No	5 from 4 people	6	92
F	No	4 from 3 people	6	80
G	No	4 from 3 people	5	78
H	No	4 from 3 people	6	75
I	Yes	3 from 3 people	5	75
J	Yes	3 from 2 people	5	92
K	No	2 from 2 people	6	66
L	No	1 from 1 person	5	65

In order for a group to share their work, at least one member of had to agree to the terms and conditions and click a button from their account. Once a group has shared their feedback it is unable to be unshared in the future. This decision was taken in order to cater for individuals who may share their work in order to view others and then unshared it again immediately after.

42% (5/12) of groups opted to share their feedback and associated code snippets. Therefore this percentage of groups had full access to the functionality provided by the system and was able to view other groups shared feedback.

Table 1 shows the distribution of the groups and whether or not the group decided to share their feedback and thereby gain access to other groups' shared feedback and source code.

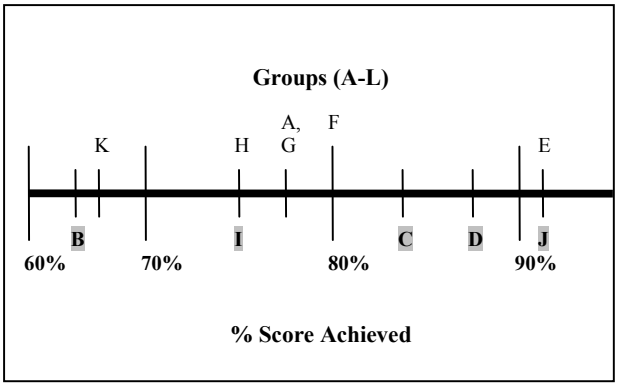

Figure 1 – Distribution of shared work

Figure 1 shows how marks for each group (A-L) were distributed and indicates which groups elected to share their work (Below the horizontal line and shaded). No generalizations can be made due

to the quantity of results generated. However, it is interesting to that a majority of groups who did not share their work occur in the mid range of marks. That is four groups between the 65% - 80% range.

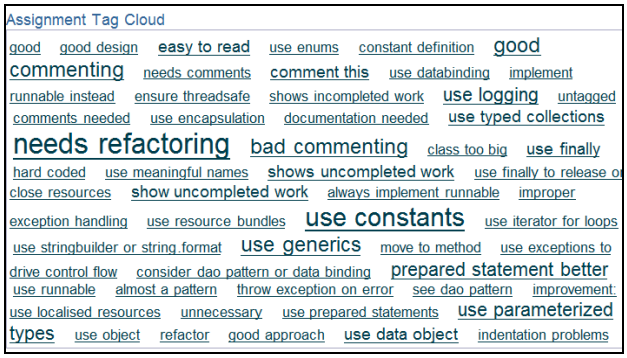

Figure 2 – Assignment Tag Cloud

In order to demonstrate the feedback students received the assignment tag cloud is shown in Figure 2. From this view it is possible to see the issues that are reoccurring in students' code. For example it is immediately clear that this group needs help refactoring from the "needs refactoring" tag in Figure 2. These types of pattern analysis may prove useful in directing future teaching.

5.2 Questionnaire Results

The response rate for the questionnaires was low, due to the time of year the study was conducted; it was during examination time. A total response rate of 21% (14/67) was achieved.

- 71% (10/14) of respondents said the feedback issued was "Very Easy" to understand, with the remaining 29% reporting that it was "Difficult". One user stated "The comments are hard to understand. Maybe they should be MORE highlighted in the code."
- 50% (7/14) of all respondents stated that the quantity of feedback received was "Very Good" or "Good", with the remainder reporting that it was "Poor".
- 93% (13/14) of respondents said that being able to see their feedback tags along side the associated source code was "Very Helpful" or "Helpful" to their learning. This supports the theory that feedback in context of the original work is more useful than if it is separated. When asked if the students could improve based on the feedback they received, 29% said yes, 29% said no and 43% were uncertain.
- 86% (12/14) of respondents said that they believed this approach to feedback would be useful when applied to individual projects.
- 36% (5/14) of respondents reported that it was useful to see other group's feedback and associated source code. This may be low due to sharing restrictions put into place by the software. For one group to gain access to other groups' feedback they must first agree to share theirs. The respondents who reported that it was not useful 64% (9/14) may have been amongst the groups who elected not to share their feedback and so were unable to view the others'.

When asked if students would opt to share their feedback and work for an individual assignment; 29% of the group said they would and 43% would not, with 29% unsure. When asked why this was the case, students who stated they wouldn't share their work suggested that their feedback is too "personal" to share. Where others in favor of sharing also had strong views; "In fact, I'd argue you shouldn't get a choice in the matter, all feedback should be shared, decouple the work from worker..." One student who said they were unsure stated "that it would depend on how well they had done on the assessment" as to whether or not they shared the work.

Some respondents explained why they were interested in seeing other groups feedback; "I wanted to see what other groups did wrong compared to us…" This suggests a competitive desire to find out which group had the better work. Another student reported that sharing feedback was "useful as to see comparison of work, and quality of feedback, plus common pitfalls…"

6. THREATS TO VALIDITY

The primary threat to this study is of the sample that has been used. For reliability, further work will need to be carried out using individual code projects. The low response rates are problematic but expected due to the timing of the research and nonetheless do give a brief insight into this approach of generating and issuing feedback.

Individuals in groups may have discussed their feedback under one user account therefore skewing the recorded results. In order to mitigate this, the participants were asked to fill out the questionnaires individually.

Since project teams are formed across friendship groups, one individual may have a friend in different group. This could circumvent the controlled sharing procedures if an individual in one group shows their feedback to an individual in another group. This will not be recorded by the software, so some relevant data may not be captured.

Students were given the summary feedback and assessment marks that they would normally get before the system was released to them. This is due to time constraints and could not be controlled. As a result students may not feel that they need to review their feedback, as they have been given it in a summarized form already. This may result in lower usage statistics than may be representative.

Since assessors have not been explicitly trained in how to use tags to assess source code there could be some confusion on the composition and generation of tags.

7. EVALUATION

This paper shows that students have mixed opinions on whether they prefer the new feedback approach to traditional mechanisms. However, a majority of students acknowledge that using feedback tags for source code has been useful to their learning.

It soon became clear from student comments that they perceived less individual relevance to the group feedback. This is common in group work as outcomes become a product of the whole group and not just of an individual. As a consequence it is likely that the way students receive and use feedback in group work is fundamentally different to that of individual assessments.

Interestingly most students agreed that the technique would be more beneficial if used with individual assignments.

The reported system access rate may be explained by another consequence of using a group project. Some groups may have used one user account to view and discuss the feedback with other members of the group present. In this case the data collected would not be a true reflection of the systems usage.

Some students suggested that they wanted to nominate the files to be assessed. The approach used for file selection was chosen for convenience and different methods will be considered in future. However, students should not be allowed to direct assessors' attention too much otherwise fundamental learning difficulties may not be uncovered.

The groups who elected to share their work represented a large spread of marks, with most of those who chose not to share feedback being the middle ranged groups. A possible explanation may be that the weaker groups were interested in how to improve their marks. The groups with the higher marks may have sought improvements to their programming style or just to see their work in comparison to their peers. Most groups who shared their feedback had a score over 80%. With courses focusing on encouraging active learning styles through assessment, it is a likely explanation that the groups who take a more active interest in theirs and others' feedback are those that receive a higher score.

8. CONCLUSIONS & FUTURE WORK

The findings of this paper serve to direct further research into how feedback tags can be used to support learning and teaching on programming courses.

Whilst the questionnaire response rates are low, the usage data provides an indication as to the usefulness of the technique. Overall, positive comments from students have outweighed the negative and this technique has been highlighted as an interesting and novel approach to assessing student programming work.

More research is required in order to determine the types of tags used by assessors when annotating source code and whether any patterns or themes can be detected.

In order to determine the applicability of this feedback approach to individual projects another investigation is being planned. This will also generate more data as to how individuals use the system in order to see their own feedback.

In addition to the model of assessor to student feedback the system can be altered to allow a peer review model as well. This will enable research on student generated tags in source code and allow comparison of student and expert feedback tags.

Further investigation into user responses using focus groups and semi structured interviews will be conducted in future research. This is in order to gain more information on student perceptions of this feedback method.

This study alone is unable to reliably measure the success of this new feedback approach. Further investigation is therefore required to quantify its effectiveness. However, it is clear that this approach for feedback generation has been useful to a majority of students in this case.

9. ACKNOWLEDGMENTS

This work is supported by the Centre for Excellence in Teaching and Learning, Active Learning in Computing.

10. REFERENCES

[1] Mason, D. & Woit, D. 1999. Providing mark-up and feedback to students with online marking. The proceedings of the thirtieth SIGCSE technical symposium on Computer science education. New Orleans, Louisiana, United States, ACM.

[2] Robins, A., Rountree, J. and Rountree, N. 2003. Learning and Teaching Programming: A Review and Discussion, Computer Science Education, 13(2): 137-172

[3] Laurillard, D. 1993. Rethinking University Teaching: a framework for the effective use of educational technology. Routledge, London

[4] Joy, M. & Luck, M. 1998. Effective electronic marking for on-line assessment. The 3rd annual conference on Integrating technology into computer science education. Dublin City Univ., Ireland, ACM.

[5] Sitthiworachart, J. & Joy, M. 2003. Web-based Peer Assessment in Learning Computer Programming. The 3 rd IEEE International Conference on Advanced Learning.