Improving On-line Assessment: an Investigation of Existing Marking Methodologies

Jon A. Preston
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
1-404-385-0026
jonp@cc.gatech.edu

Russell Shackelford
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
1-404-894-9217
russ@cc.gatech.edu

ABSTRACT

We are in the process of developing an on-line marking system for use in our large-scale CS1 and CS2 courses. To better accommodate the needs of our numerous raters, we investigated their current methodologies in marking students' work; this paper presents our findings from recording "think out loud" marking sessions and surveys. A prototype for an on-line assessment software tool is described; this tool allows markers to view students' work at various degrees of detail ranging from complete, low-level to "meta-level." We believe such a system is beneficial for improving the marking process in large-scale and distance classes.

Keywords

Educational technology, assessment, on-line marking, evaluation.

1. DEFINITIONS

For the purpose of clarity in this paper, we define *marking* student work as the process of noticing errors, providing corrections, and assigning numerical grades. We will use the terms *TA* and *rater* to denote an assistant to the instructor who evaluates student work.

2. MOTIVATION

Accurate and meaningful assessment is vitally important for many reasons. First, it provides meaningful feedback to students and instructors; quality assessment informs students of their mistakes and successes and informs instructors of student knowledge. Second, it establishes confidence in the measurement of student performance; without accurate assessment, neither students nor instructors have a reasonable gauge of student knowledge. Third, it provides instructors and administrators with the ability perform quality control; collecting reliable performance data enables examination of the instructional process for courses. Finally, accurate assessment makes new educational research opportunities possible; customized courses, better use of class time, and student performance trend analysis are a few examples of such possibilities [4].

Inter-rater reliability is a serious problem that undermines the consistency and quality of assessment. Specifically, we have found in previous studies that there is a real problem with inter-rater reliability in large classes [6]. This problem is exacerbated in our CS1 and CS2 courses where hundreds of students enroll each term and over one hundred teaching assistants mark over 4000 assignments each week. In addition, because of such massive class sizes, students and instructors often do not have close contact and interactions. As a result, it is extremely important that the students get detailed and constructive feedback on their graded assignments from the teaching assistants [5,6,7].

3. BACKGROUND

Our previous work involves a repository of assignments and student work. This system manages many of the tasks involved in managing a large-scale course such as assignment distribution and collection, newsgroup interactions, and grade calculation [1,7]. A dynamic survey tool that allows for customized grading sessions based upon student work is a part of this system [6,7]. This on-line marking tool is useful in querying raters about the quality and correctness of students' work. Until now, the on-line presentation of the student work has not been investigated. This component of the on-line marking system has been deferred to paper or a simple text editor.

Figure 1 shows the entire system and how instructors, students, and raters interact with the various components. Notice that the on-line marking tool is composed of two separate windows which interact via message passing; the viewer allows for in-text annotation and the criteria interface saves the state of the viewer and provides for dynamic querying of the students' work [5, 6].

While this on-line marking system has improved in the last three years, we found that raters were still reluctant to use the system and completely accept it. While some of this non-acceptance can be attributed to "new system syndrome," we wanted to find out why raters weren't more accepting of the on-line marking tool.

Consequently, the purpose of this study is to investigate the current methodologies employed by raters in marking students

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ITiCSE '99 6/99 Cracow, Poland
© 1999 ACM 1-58113-087-2/99/0005...$5.00

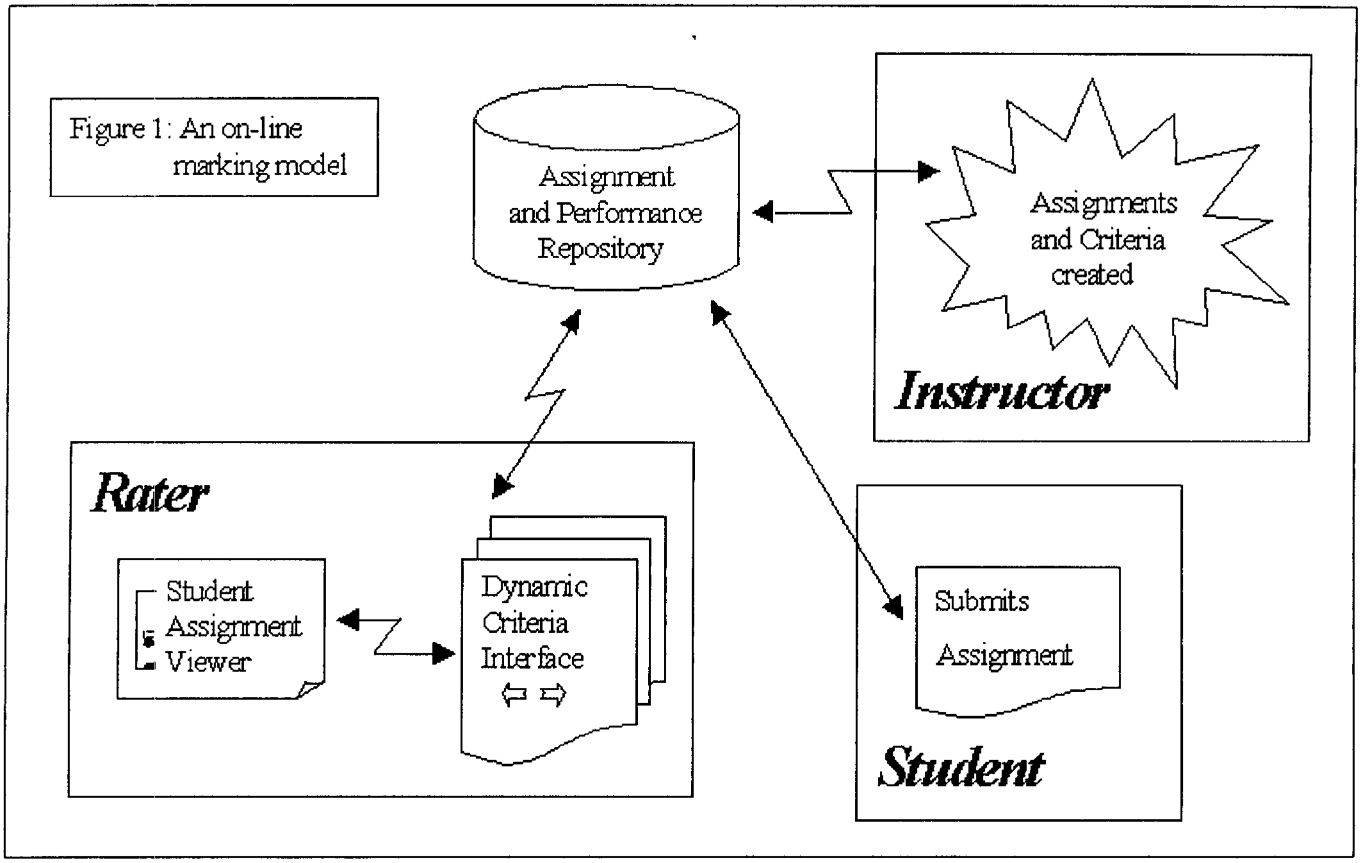


Figure 1: An on-line marking model

Assignment and Performance Repository
Assignments and Criteria created
Instructor
Rater
Student Assignment Viewer
Dynamic Criteria Interface
Submits Assignment
Student

work. After the investigation, we develop a tool that supports the raters marking activity by better modeling their existing marking methodologies.

4. THE INVESTIGATION

We began by examining the current marking methodologies of the raters in our CS1 and CS2 courses. This phase of the research involves two activities.

4.1 Marking "Out loud"

Six TAs voluntarily participated in this phase of the study; their experience ranged from a first-term TA to two veteran TAs each with over six terms of prior marking experience.

The TAs marked their own students' assignments and were asked to mark the assignment as they normally would. The TAs were also asked to "think out loud" and describe what they were thinking as they marked the assignment. If a TA paused for more than two seconds while grading, we prompted them to describe what they were doing or thinking.

Each grading session involved one TA grading one assignment that was randomly selected from their students. Each of the eight sessions was recorded on audio tape and then transcribed. We then performed phenomenological qualitative analysis on the transcribed sessions and copies of the graded assignments.

Sessions lasted from 15.2 to 34.2 minutes (average of 22.3), with the variance depending largely upon the rater rather than the assignment.

4.2 The Survey

In this phase of the study, raters were asked:

> *Please describe (in as much detail as possible) the process/methodology you use in grading assignments.*

Seven responses were collected from TAs, all of whom had not participated in the first phase of the study.

Respondents report using various support tools to make marking more efficient. One marker uses scripts and macros to facilitate his on-line marking. Others report running scripts to automatically compile and run test input through students' programs. When the programs would not compile, TAs report that they fix small syntactic errors (missing parenthesis and semicolons, etc.) and evaluate the programs by running them interactively. Note that our CS1 course uses a non-compilable language (pseudo-code) and our CS2 course uses a compilable language; thus the methods and tools used in evaluating the student work vary between the courses.

Raters also report that they first examine the question that is asked of students in order to build a mental model of how they (the TAs) would solve the problem.

In addition, TAs report that they examine the students' work before looking to the criteria. Obtaining this "big picture" view of the work tells the rater "if ... [the student] truly understood what they tried to implement."

5. CONCLUSIONS

While each rater has his own distinctive marking style and methodology, we can draw some important conclusions by examining the grading sessions and the survey responses.

First, it is clear that all raters begin the marking session by forming their own mental model of the question and the correct answer. TAs examine the question given to the students and then determine how they (the TA) would solve the problem. After this, the TA begins to examine the student work.

After understanding the question and constructing a solution, TAs examine the students' answers before looking to the criteria. TAs focus on the answer given and apply the criteria to the students' answer rather than focus on the criteria and apply the answers to the criteria.

This approach is quite different than what we expected. Our previous work focuses on the presentation of criteria to improve the marking process. By examining how raters mark, we find that they focus on the student work; the criteria takes lower precedence and is "molded" to fit the work being assessed.

In general, raters examined the students' work at a high level before delving into the details of implementation. TAs examined the module headers and contracts (purpose, pre- and post-condition comments) for the problems in an effort to get the "big picture" of the students' solutions. The low-level implementation details were later examined for correctness. This implies that raters take multiple views of the students' work – ranging from a high-level (meta-level) examination of the details in the code to a low-level inquiry. We believe TAs use this meta-level view to query student effort and conceptual correctness for the given task. We also note that TAs went back and forth between these high- and low-views while evaluating the students' work.

Our courses use "feedback codes" to classify student errors into categories; these categories are later used for statistical analysis on the types of errors students are making in the courses and to give students insight into the types of errors they make. In this study, we note that no TA had a comment code list with them while participating in the "mark out loud" grading sessions. This supports our previous findings that raters remember and use a small subset of all available (and appropriate) feedback codes while marking. Consequently, the on-line marking should abstract the details of these comment codes, embedding them in the internals of the criteria where the rater does not have to remember them.

It is important to note that our study focused on programming assignments in a CS1 course, but the on-line marking model and associated software is applicable in assessing any assignment in various courses ranging from English to Computer Science.

6. THE PROTOTYPE

After examining the information gathered concerning marking methodologies currently used by our raters, we believe any on-line marking system should:

- Place emphasis on the students' submission rather than the criteria
- Allow for a big-picture view of the students' work, hiding and displaying implementation details when needed
- Allow for quick and easy navigation between sections/problems of the work
- Highlight language syntax to improve readability
- Enable easy annotation features [8]
- Allow for "by-problem" or "by-student" marking
- Separate the interface of assessment from the implementation (ie. hide the points and feedback codes from the rater)
- Automate point/grade submission and downloading and uploading of needed files [2]

Figure 2 shows an initial prototype of the user interface for on-line marking. It allows the rater to "close" sections of the code/answer and view a high-level picture of the work. These sections can also be "opened" and even evaluated to check correctness of the low-level details.

7. ISSUES AND NEW QUESTIONS

The on-line marking system described above is a work in progress. We continue to learn and perfect the system to better support raters in assessing students' work. Certainly there are still open issues and questions.

The most important issue that is still unresolved is the degree of control that raters have while using the on-line system. With the pen and paper approach to marking, raters interpreted the criteria given to them and have complete control over points and feedback codes. While the criteria imposes structure, raters often do not follow the criteria; this is evidenced in our previous studies [6] and in our analysis of the marking sessions in this study.

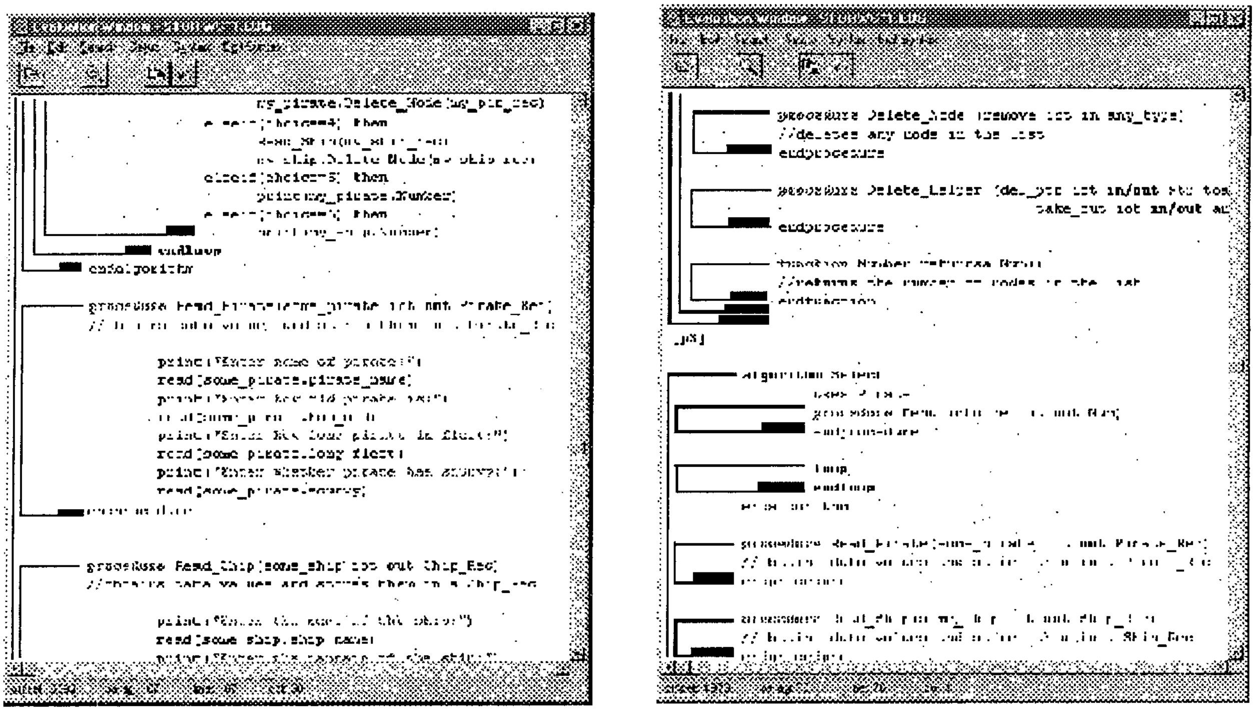
The on-line marking system hides the implementation details of assessment, embedding points and feedback codes in the internals of the system and away from the raters. While this is advantageous to improving the consistency and quality of the assessment, there are some problems that this raises.

First, raters may feel disenfranchised and somewhat "powerless" while assessing students' work. Certainly the system is different, so steps must be taken to overcome "new system syndrome," but more importantly, raters' view of assessment must change. No longer is assessment about distributing points; now assessment can focus more on the examination of the quality and correctness of students' work.

Certainly the on-line marking system must be flexible and allow markers to annotate students' work. The system should bolster the marking process, and we have made every effort to model the tools after the current marking methodology.

In addition, there will always be cases in which the criteria provided in the on-line marking system does not apply to the students' work. Examples of this situation abound in practice due to the limited foresight of those who create the criteria (ie. it's impossible to foresee all possible solutions to a given problem). A feasible solution to this problem is absolutely necessary in order for the on-line marking system to be usable.

Currently, the rater simply uses his judgement, notes errors and distributes points in this situation. Unfortunately, there is no effective, scalable mechanism by which these cases and errors in the criteria are reported to the appropriate personnel when using a pen-and-paper based marking methodology.



```
                    my_pirate.Delete_Node(my_pir_rec)
            elseif(choice=4) then
                    Read_Ship(my_ship_rec)
                    my_ship.Delete_Node(my_ship_rec)
            elseif(choice=5) then
                    print(my_pirate.Number)
            elseif(choice=6) then
                    [illegible]
    endloop
endalgorithm

procedure Read_Pirate([illegible] Pirate_Rec)
// [illegible]

        print("Enter name of pirate:")
        read(some_pirate.pirate_name)
        [illegible]
        [illegible]
        print("Enter the loot pirate in fleet:")
        read(some pirate.loot fleet)
        print("Enter whether pirate has scurvy:")
        read(some_pirate.scurvy)
[illegible]

procedure Read_Ship(some_ship isoft out Ship_Rec)
//[illegible] Ship_rec

        print("Enter the name of the ship:")
        read(some ship.ship name)
        [illegible]
```

```
    procedure Delete_Node (remove isoft in any_type)
    //deletes any node in the list
    endprocedure

    procedure Delete_Helper (del_ptr isoft in/out ptr to[illegible]
                                    take_out isoft in/out an[illegible]
    endprocedure

    function Number returnsa Num
    //returns the number of nodes in the list
    endfunction
[illegible]

algorithm Select
        [illegible]
        [illegible]
        [illegible]

        loop
        endloop
[illegible]

procedure Read_Pirate([illegible] Pirate_Rec)
// [illegible]
[illegible]

procedure Read_Ship([illegible] Ship_Rec)
// [illegible] Ship_Rec
[illegible]
```

Figure 2: Different Views of a Student's Work

The new system will still have this problem of non-applicable criteria. But the on-line marking system contains an automatic feedback by which raters can send the special cases and problems with the criteria to the person responsible for maintaining and updating the criteria.

Unfortunately, since points and feedback codes are not visible to the rater, it is much harder to assess these "different" cases. This is quite a difficult problem to solve, as these "special cases" always pose the greatest problems to any system. We will further investigate the appropriate solution is to this issue.

A related topic of interest is how much control over points and feedback codes should be given to raters. A possible solution to this issue is to have a portion of the grade for the assignment devoted to raters' discretion. This would allow raters to assess intangibles in the students' work such as style.

8. ACKNOWLEDGEMENTS

We sincerely thank all the teaching assistants that donated their time and input to this study.

9. REFERENCES

[1] Canup, M and Shackelford, R. Using Software to Solve Problems in Large Computing Courses. *SIGSCE Bulletin: '98 Technical Symposium* (Atlanta, Georgia) ACM, New York. March, 1998. 135-139.

[2] Dawson-Howe K. Automatic Submission and Administration of Programming Assignments. *SIGSCE Bulletin*, 27, 4 (December 1995), 51-53.

[3] East, Philip J. Including Quality Assessment in Programming Instruction. In *Conference Proceedings of National Educational Computing Conference '98* (San Diego, CA) ISTE, Eugene, OR. June 1998.

[4] Hopkins, Kenneth D. *Educational and Psychological Measurement and Evaluation.* Allyn & Bacon, Boston, 1998. 1-15.

[5] Nicholson, A et al. Computer Assisted Assessment. Summary from the 1994 TLTP Conference held at Cardiff. icbl.hw.ac.uk/tltp/conferences/cardiff94.html

[6] Preston, J. Evaluation software: improving consistency and reliability of performance rating. In *Proceedings of ITiCSE* '97 (Uppsala, Sweden) ACM, New York. June 1997.

[7] Preston, J. and Shackelford, R. A System for Improving Distance and Large-Scale Classes. In *Proceedings of ITiCSE* '98 (Dublin, Ireland) ACM, New York. August 1998.

[8] Price, B. and Petre, M. Teaching Programming through Paperless Assignments: an empirical evaluation of instructor feedback. In *Proceedings of ITiCSE* '97 (Uppsala, Sweden) ACM, New York. June 1997, 94-98.

[9] Toothman, B and Shackelford, R. The Effects of Partially-Individualized Assignments on Subsequent Student Performance. *SIGSCE Bulletin: '98 Technical Symposium* (Atlanta, Georgia) ACM, New York. March, 1998. 287-291.