We regret to inform you that your submission # 208

Title: Online Programming System for Code Analysis and Activity Tracking

has NOT been accepted for inclusion in the SIGCSE 2017 program. Reviews are appended at the

end of this email.

For SIGCSE 2017, we accepted 105/350 papers (30%), 16/29 panels (55%), 12/22 (55%) special

sessions, and 36/45 workshops (80%). For comparison, the SIGCSE 2016 numbers were: 105/297

papers (35%), 12/22 panels (55%), 10/17 special sessions (59%), and 36/72 (50%) workshops.

This represents a 17.8% increase in paper submissions, but a significant drop in workshop

submissions.

In keeping with SIGCSE’s rigorous standards, each submission was carefully reviewed by at least 5

members of the Program Committee (PC) and reviews were provided to the APC/Senior PC for

papers, and to the respective Workshop/Panel/Special Session Track chair(s). First, submissions

were assigned to at least 5 members of the PC based on their expertise, interests, and other

factors. Each submission received at least 5 reviews, and some received 7 reviews. After the initial

reviews were submitted, the APC/Senior PC and Track Chairs facilitated discussion amongst

reviewers in order to resolve differences and correct misunderstandings. The APC/Track Chairs

were then encouraged to submit a summative recommendation to the Program Chairs. Our final

decisions were based on reviewer scores, APC/Track Chair recommendations, as well as space,

novelty, and timeliness considerations.

Below, you will find the reviews. We hope you find the feedback from the reviewers helpful and

thank you again for your submission to SIGCSE 2017. We hope you still plan to attend SIGCSE

2017. Be sure to register and book your hotel early as the room blocks often fill up! The tentative

SIGCSE program will be available by October 18 at the SIGCSE 2017 website

(<http://sigcse2017.sigcse.org/>).

We value your work and your contributions to SIGCSE, and hope to see you in Seattle from March

8-11.

Sincerely,

Tiffany Barnes (NC State) and Dan Garcia (UC Berkeley), Program Chairs for SIGCSE 2017

[sigcse2017-program@cs.vt.edu](mailto:sigcse2017-program@cs.vt.edu)

----------------------- REVIEW 1 ---------------------

PAPER: 208

TITLE: Online Programming System for Code Analysis and Activity Tracking

AUTHORS: Tian Qiu, Mengshi Feng, Sitian Lu, Zhuofan Li, Yudi Wu, Kaiwen Yu, John Laiman, Nathan Chow, David Nelson, Constantine Roros and Yung-Hsiang Lu

OVERALL EVALUATION: 3 (Marginal Tend to Reject: Not as badly flawed; major effort necessary to make acceptable but content well-covered in literature already)

----------- Summary -----------

The authors designed an online IDE and used it to track errors students made as they programmed.

----------- Strengths -----------

The collection and classification of student errors made during development of programs is the strength of this paper. The authors put significant (!!) effort into building the online IDE and are justifiably proud of it, but I do not believe that is the strength of this paper.

----------- OVERALL EVALUATION -----------

The data collected on errors during development is interesting. However, they do not do the minimal analysis of the data such as Table 2 in the Experimental Results section to see if the differences are significant.

----------------------- REVIEW 2 ---------------------

PAPER: 208

TITLE: Online Programming System for Code Analysis and Activity Tracking

AUTHORS: Tian Qiu, Mengshi Feng, Sitian Lu, Zhuofan Li, Yudi Wu, Kaiwen Yu, John Laiman, Nathan Chow, David Nelson, Constantine Roros and Yung-Hsiang Lu

OVERALL EVALUATION: 3 (Marginal Tend to Reject: Not as badly flawed; major effort necessary to make acceptable but content well-covered in literature already)

----------- Summary -----------

The paper presents information about an online programming system for code submission, compilation, testing, analysis, and activity tracking.

Audience: several educators might be interested in learning about a new system that is somewhat different from systems.

Main points: the design of an online programming system is presented and some results of a small course (N=42) are discussed, which indicate that the system can be helpful.

----------- Strengths -----------

The topic itself is of interest, as helpful online programming systems that combine safe operation (here, using Docker) and automatic testing and grading can be a large help in larger classes.

The system is described with sufficient detail to get an understanding, if not enough to build another one like it.

An experiment with N=42 was run and in many cases, the experimental group ended up with a better score than the control group.

Some analysis on where the system might have been less of a help, perhaps even a hindrance, is also done.

----------- OVERALL EVALUATION -----------

The underlying idea and the system design seem to be sound and workable.

My main points of improvement for the paper are as follows:

- please proof-read the English. There are several small mistakes (including missing or superfluous commas, missing articles ("a", "an" or "the"), and incorrect genitive forms. Also, the hyphenation in Table 1 is strange, for example "Proposed-System" should be "Proposed System" (no "-").

- I would have preferred to see more in-depth presentation about the system and/or sample code problems.

- The evaluation should also use some figures to give a quick idea of where the experimental group "did better", so readers do not have to extract this information from the text itself.

- The paper does not follow the author guidelines concerning the format. There is no (anonymized) area for author information, incomplete metadata, no space left for the copyright notice, and the line spread is incorrect ("regular" papers are far tighter). As such, once the adaptations are made, there will be plenty space for additional content including the one I was missing, as mentioned above.

----------------------- REVIEW 3 ---------------------

PAPER: 208

TITLE: Online Programming System for Code Analysis and Activity Tracking

AUTHORS: Tian Qiu, Mengshi Feng, Sitian Lu, Zhuofan Li, Yudi Wu, Kaiwen Yu, John Laiman, Nathan Chow, David Nelson, Constantine Roros and Yung-Hsiang Lu

OVERALL EVALUATION: 3 (Marginal Tend to Reject: Not as badly flawed; major effort necessary to make acceptable but content well-covered in literature already)

----------- Summary -----------

A Web Integrated Development Environment (WIDE) tool is described that, in addition to providing the features of an IDE, tracks student errors and provides a comprehensive set of features that are also present in other assorted similar WIDE systems. By providing data about stduent errors and their analysis to the instructors, the tool provides instructors valuable information about the students' preparation, and learning for a sophomore level C programming course.

The intended audience for this submission are instructors who also teach the C programming language and are interested in obtaining some additional information about the student's programming learning process.

In the single, summer-course based study, the claim is made that the tool provodes udseful information to the instructor and results in some improvement in student learning.

----------- Strengths -----------

The tool attempts to use the current WIDE-based technologies to assist in student programming and error analysis. The submission claims that the tool and the data provided by it can result in improved student learning for programming in C.

----------- OVERALL EVALUATION -----------

The idea of such tools has been in existence for many years now. While the submission provides a quick survey of the current WIDE tools, it lacks in a comprehensive review of prior work in this area (of detecting and tracking student programming errors) by several researchers (and widely reported in past SIGCSE and ITiCSE conferences).

Some other issues that lead to the rating:

1. It has not been suffieciently tried and tested. The results reported are encouraging, but not affirming. A sample size of 21 does not give any reasons for making sweeping generalizations about the efficacy of the tool.

2. The tool is for students learning the C programming language. Students in the test already had a prior course in the C language. Why wasn't the study performed in the course where tsudents were learning C?

3. The study was carried out in a summer course, not a regular semester-long course on TWO assignments. This is a one-of study and probably not replicable in a semester long course, or in several assignments. The author(s) needs to actually carry out the more longitudinal study and then report back, if it shows encouraging results.

4. It is not clear whether this tool will be suitable for other (more complex) programming languages that are more prevalent in CS curricula today.

5. The analogy with pilot training and professional software devlopment training is not applicable for sophomore level students who are still learning concepts in computing and programming.

----------------------- REVIEW 4 ---------------------

PAPER: 208

TITLE: Online Programming System for Code Analysis and Activity Tracking

AUTHORS: Tian Qiu, Mengshi Feng, Sitian Lu, Zhuofan Li, Yudi Wu, Kaiwen Yu, John Laiman, Nathan Chow, David Nelson, Constantine Roros and Yung-Hsiang Lu

OVERALL EVALUATION: 3 (Marginal Tend to Reject: Not as badly flawed; major effort necessary to make acceptable but content well-covered in literature already)

----------- Summary -----------

The paper presents the design of a code analysis system used to support teaching of sophomore C programming class and discusses the impact it had when used with a group of 42 students. The effectiveness is evaluated by considering student performance on an end of course multiple choice quiz comparing the performance of students who used the tool for a section the course examined by the quiz with those who didn't.

The audience would be those delivering programming courses. Its main points are the design of the tool and an insight into its use within a course.

----------- Strengths -----------

The paper presents a detailed design of the tool and presents some interesting findings in terms of the types of issues where students appear to have benefited from using the tool.

----------- OVERALL EVALUATION -----------

It would be useful to introduce the types of assignment the tool was used for in each of the homework assignments. Also some details of what the comparison group were provided with in terms of support when not using the tool. It would clarify the contribution of the tool. It is not clear if the students in the experimental group only had access to the tool or whether they had access to the usual supports plus the tool.

I'm not sure that the detail of the technical design of the tool is particularly relevant, it would be more useful to see it from the perspective of course design, assessment design, learning theory etc.

----------------------- REVIEW 5 ---------------------

PAPER: 208

TITLE: Online Programming System for Code Analysis and Activity Tracking

AUTHORS: Tian Qiu, Mengshi Feng, Sitian Lu, Zhuofan Li, Yudi Wu, Kaiwen Yu, John Laiman, Nathan Chow, David Nelson, Constantine Roros and Yung-Hsiang Lu

OVERALL EVALUATION: 2 (Probable Reject: Basic flaws in content or presentation or very poorly written)

----------- Summary -----------

Authors have developed an online programming environment that provides feedback for beginning programmers, without the added work of those programmers learning to handle and interpret the software that generates that feedback. So, for instance, a student's submission might provide the student with notes on that submission's performance, while logging any errors made along the way for future perusal by the professor. Primary audience appears to be teachers of introductory programming courses.

----------- Strengths -----------

Paper provides a compare-and-contrast discussion of a number of other existing alternatives, which is helpful for evaluating the merits of this new entry. The paper also includes experimental results, which are useful for evaluating the merits of the system.

----------- OVERALL EVALUATION -----------

I found this paper difficult to read, on a couple of different axes. In no particular order:

1) The grammatical editing is, while not terrible, bad enough to be distracting. Word substitutions ("stimulate" for "simulate"), subject-verb disagreements ("Memory leak does not cause..."), and missing words (Same example) are relatively common throughout.

2) The paper has a critical organizational problem. There appears to be no section of text that actually lays out exactly what the program shows a student; that information might be expected in Section 3, which appears to be a summary of functionality, but other bits of explanation are laid out in Section 2 (Related Works) and as late as section 4.5 (System Usage). This makes it very difficult to gain any kind of comprehensive sense of the benefits provided. I would suggest reorganizing the paper to provide a single descriptive section, compiling all benefits for student and professor - but regardless of the specifics, some radical reorganization is needed.

3) The lack of organization also make some of the commentary difficult to assess. I found section 4.4 (Posttest) very difficult to follow; for instance, what does it mean that "The average advantage was 7%?" Does this mean that the experimental group averaged 7% better on all 12 type-3 questions, or only on the 8 of the 12 in which they scored higher? If the former, how did they perform on the other questions? From the same section, when you say that the scores "of the two groups" were 13.9 and 13.95, what are "the two groups" - the experimental and non-experimental groups? Does that imply that the non-experimental group outperformed the experimental group? Perhaps that's not at all what you intended to say, but these paragraphs were baffling throughout.

It would also be nice to see p-values on some of these results. How significant are the differences, given your small sample sizes? In particular, the failure of the experimental group on HW 5 seems very notable, bringing the average homework scores for the two groups almost back to equality; how likely is it that the variation on the other assignments is just random noise? I also have trouble following your explanation of this low score; you say that the experimental students had grown dependent on the system, but I thought students in both groups alternated between using the system and not - why should this group, alone, become so dependent?

4) To some extent, I'm unimpressed by the problem to be solved here. Your opening metaphor suggests that configuring an IDE is a highly technical task, comparable to airplane maintenance, but it's a task I regularly see freshmen with no background in CS accomplish. Again, I struggle somewhat to see exactly what benefits your system is providing due to the organization, but a fair number of them seem to be already covered by a modern IDE, and many others (performance, for instance) seem of little concern to a beginning programmer.

------------------------- METAREVIEW ------------------------

PAPER: 208

TITLE: Online Programming System for Code Analysis and Activity Tracking

The paper presents a web-based IDE for the C programming language, and its use in an experiment. Strengths include the novel approach with comparison/contrast to some previously existing tools, and analysis of the limits of system efficacy. The paper could be improved by a more comprehensive review of the prior work, and further experiments. Additionally, writing quality could be improved, as well as the organization of the paper for increased reader understanding of the benefits of the system over existing IDEs.

--