

InClass Assistant, Enhancing Student Class Participation

Yolanda Martínez-Treviño
Computer Science Department
Tecnologico de Monterrey, Campus Monterrey
Monterrey, Mexico

Abstract—Work in Progress. The first programming course has never been an easy task for students; on the other hand, it is known that students' participation in classroom activities benefits their learning process. Our approach is to increase students' participation in the classroom as well as to make it easier and faster for the professor to provide feedback in the classroom activities, so that students can know if their participation was correct in a short period of time. To make this possible, we designed *InClass Assistant*, a software tool that promotes class participation and facilitates the way in which the professor gives feedback on class activities. In this paper, we present the tool and the way it has been used in our courses. We also show the students' and the professor's points of view about using this tool in class sessions.

Keywords— *Technology in the classroom; class participation; programming; in-class feedback, CS1/2*

I. INTRODUCTION

The first programming course has been an obstacle for many students around the world [1]. Guzdial [2] analyzed research done in three universities and two countries and found that even though most of the students passed the Computer Science I (CS1) courses, they should have failed them because they did not have what he called "passable performance".

On the other hand, different studies have shown that using active learning strategies in the classroom results in a lot of advantages for the students, including an increase in their levels of comfort and confidence and improvement in the retention of the material. [3,4]

The author of this work has been using active learning activities in her courses for 13 years [5]. In these activities, students are asked to program on their laptops the solution to the problems the professor assigns. While students work individually or in pairs, the professor walks around to answer students' questions.

In this model, the communication between professor and students is not necessarily the best because not all of the students get feedback from the professor; on the one hand, some students prefer do not ask; on the other hand, the professor lacks the time to go with all of the students. In this kind of sessions, the professor has observed that some students do not work on the program assigned, or that some students feel embarrassed because they do not know how to develop the program, and thus, do not ask anything.

In this paper, we present a tool designed to increase the participation in active learning class sessions and to make feedback for classroom activities easier and faster. As a consequence, it is expected that the students' learning improves.

II. RELATED WORK

Ubiquitous Presenter [6] is a classroom communication tool. It enables students to view the slides and ink comments presented by the instructor. Students can also submit their answers adding text or ink comments to the professor's slides. In 2007, Ubiquitous Presenter was used in a CS1 class by the author of this paper and it was highly accepted by the students. They could send their solutions and receive feedback. In that occasion, some students mentioned that they thought they were the only ones with problems to learn to program, and through the use of this tool they realized that there were more classmates with the same problems. Other students commented that they could learn when they watched how the professor corrected others' errors. Students also mentioned they liked that the submissions were anonymous because other people would not know who had made the mistakes [7]. However, this tool was designed to share digital ink, which is not the best way to share programming code.

PeerPresents [8] is a software tool designed to encourage peer review during class sessions while students are giving a presentation. The underlying principle is that giving and receiving feedback is a key activity for students to learn, specially for mid-sized to large groups where the time needed to receive feedback from peers is very short in comparison with receiving feedback only from the instructor. In this tool the students and/or professor define a rubric before the presentation, and during class time, peers respond to the rubric questions and vote on peers' responses allowing the presenter to have a good amount of feedback at the end of the session.

MarkOn [9] is a tool intended to encourage students to ask questions during class time; it promotes students' participation because it is known that students' active participation in class has shown to have benefits. The underlying belief is that students keep the questions to themselves due to the psychological stress that it represents. Therefore, with this tool, students can post questions during class time, and the instructor is notified of those questions through a software tool; then, the instructor either stops to answer, or goes on with the lecture for a while and then responds to the questions.

We also found that blogs, wikis [10] and collaboration tools [11] have been used to support the learning process in different topics, for which the whole group collaborates to come up with a solution.

All the above mentioned tools are meant to encourage class participation since the faculty agrees that the more the students participate, the better their learning.

Nevertheless, the focus of our tool is different. We focus students' participation on solving programming problems. Moreover, we want them to do this individually since we need to ensure that every single student develops the competencies needed. Therefore, we introduce a tool not only designed to promote students' participation in class, whether individually or in pairs, but also one that does not allow to share solutions with classmates and that facilitates the instructor to give feedback on class activities.

III. THE TOOL

InClass Assistant is a software tool designed to support class participation, increase the professor-student in-class communication and improve providing feedback on class activities.

This tool enables the professor to define one or several activities, post a comment describing the activity; and then receive comments from students with individual or teams' solutions to the problems.

The professor can see the students' answers in the moment they are sent on a blog-like interface, and then she can give immediate feedback out loud or approach the student individually.

As we said before, the interface of InClass Assistant is similar to a blog but has two significant differences. The first one is that only the professor can see the solutions students upload. Therefore, each student or team has the opportunity to define their own unbiased solution. The second difference is that the tool is designed to share programming code; which means that it keeps all the programming formats, and it has syntax coloring for C++, Java, C#, and Python languages.

InClass Assistant is not meant to support compilation or execution; its sole purpose is to share code.

InClass Assistant gives the professor the option to mark a submission as the solution for the activity; in this case, the tool uses a different color so that students can identify the professor's solutions easily.

The tool can keep all the students' and professor's submissions for the course term. In this way, students can always have the examples and solutions given by the professor as well as their own answers.

Figure 1 shows a screenshot of InClass Assistant. In the bottom is the description of the activity sent by the professor; in the middle is a solution sent by a student, and in the top is the solution to the problem sent by the professor. The tool uses different colors to indicate who sent the submission.

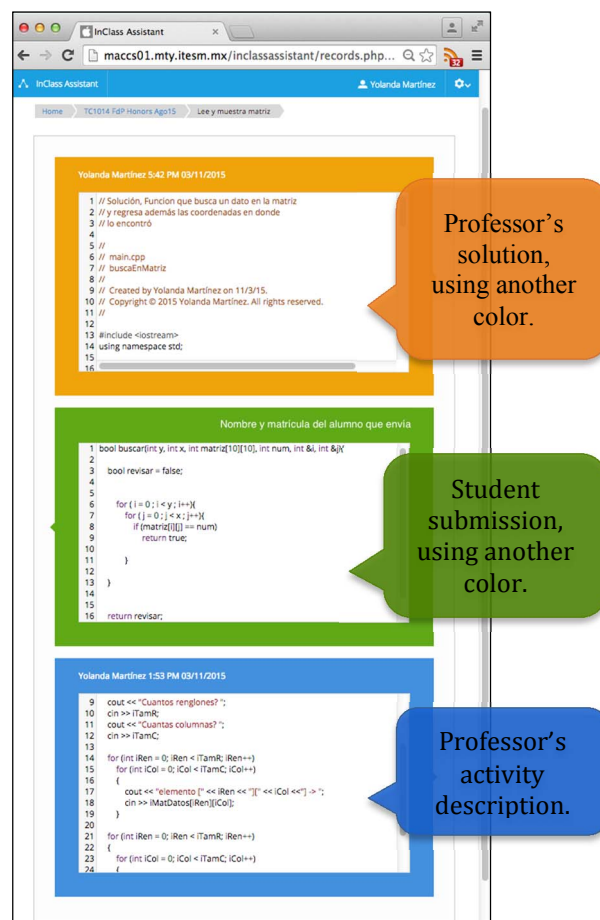


Fig. 1. Screenshot of InClass Assistant

The header of each submission shows the student's ID, the student's name, the date and time of submission. To protect the student's identity, this header can be hidden when the professor projects the screen.

IV. USING THE TOOL

InClass Assistant has been used in classes conducted with an active learning approach. A typical class is a 90-minute period which includes a 15-20 minutes' explanation and Q&A at the beginning of the period. The rest of the session is for students to work on exercises. The exercises are usually programming exercises that students have to do on their laptops, either individually or in pairs.

During the 2016 spring term, the tool was used in different ways as described in the following sections:

A. Solving an exercise and displaying solutions

The professor asked the students to do an exercise; after a while she displayed what everybody had sent. Then, she asked the group to decide if the solution presented was correct or not; if the answer was wrong, students were expected to find the errors or to make suggestions for improvement.

V. THE TEST PHASE

The professor used this type of sessions to make it evident that there were different ways of solving the problem and to show the common errors in the solutions.

B. Solving an exercise and giving feedback

This is a variation of the previous type of activity. In this case, the professor was the only person who could see all the submissions and gave feedback out loud to each person from her desk.

The students who had correctly completed the exercise could continue with the following exercise, described in another activity in the tool. On the other hand, those who had problems received feedback and kept working on the problem. They eventually received feedback.

In this type of activity, the professor identified and explained to the group the common errors. Although it was difficult to identify all the students having problems during the class session, the tool made it easier for the professor to find people who had not sent the exercise or who had difficulties and tried to help them.

Usually, there was not enough time to talk to all the students that had questions; however, giving feedback out loud to the whole group was useful to students who had similar mistakes.

C. Completing a programming exercise

The professor defined an exercise, which consisted on completing a piece of code. The professor shared the incomplete piece of code through InClass Assistant, so that everybody could copy this code and complete the program.

This way of working was very agile because thanks to InClass Assistant's blog-like interface, students just had to copy and paste the piece of code and concentrate on completing the exercise, avoiding the waste of time needed to download files.

D. Sharing examples

When the professor built a solution with the group's participation, she could easily share it posting the code as a solution. This improved the professor-student communication because students could then look for the title of the activity and find the solution easily.

Sometimes it was also useful for the professor to modify the exercise to solve it some other way; when this happened, she could just share it again. This way of working allowed the professor to share several examples in very short time. Before the use of InClass Assistant, the exercise had to be saved with different file names each time and all versions were uploaded at the end of the session.

E. Giving extra credit

Another way of using this tool was to give extra credit to the first n students who answered a given exercise correctly. This was easily done with InClass Assistant because the tool registered the date and time each submission was made, making it fair and easy to give credit to the first students who gave a correct answer.

InClass Assistant was used for the first time in three different groups in the Fall Semester of 2015. One regular Computer Science 1 (CS1) group with 35 students, one honors program CS1 group with 19 students, and one Object Oriented Programming non-Computer Science major group with 21 students (business and digital arts students).

During that semester the tool was used in approximately half of the class sessions; therefore, it was possible to ask students to compare class sessions with and without the use of the tool at the end of the term.

We administered two short surveys to the students. The first one - with only one open question to obtain students' comments- was given in the middle of the course; the other, with several questions, was given at the end of the course. This last survey was answered by 66 students. At the end of the course, we also conducted a focus group to get students' feedback.

A. Students' opinion

In this section, we present some of the responses obtained in the surveys.

We asked the students "What do you prefer, a class where InClass Assistant is used or where it is not used?" Figure 2 shows that most participants (67%) preferred a class where InClass Assistant was used.

Table 1 shows the percentage of people answering other statements. Sixty-four percent of the students agreed or strongly agreed with the statement "I participate more in class when InClass Assistant is used." Seventy-three percent agreed or strongly agreed with the statement "When InClass Assistant is used, I receive feedback during class time." Eighty-six percent agreed or strongly agreed with the statement "Send/Receive code is easier and faster than with other tools." In this case the other tools were email or the official tool used in the course which implied from logging in, to looking for the right place to add a file, to uploading it, to having students go to this same place to download it. Finally, seventy-six percent agreed or strongly agreed with the statement "Seeing other people's code helped me to learn better."

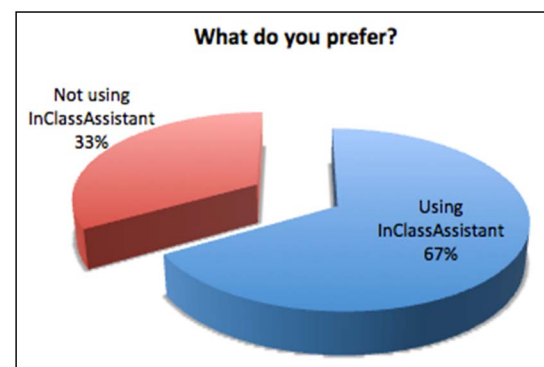


Fig. 2. Chart with the responses to the question "What do you prefer?"

Table 1. Percentages of students answering each option.

	strongly agree	agree	Neither Agree nor Disagree	Disagree	Strongly Disagree	Did not answered
I participate more in class when InClass Assistant is used	23%	41%	24%	11%	1%	0%
When InClass Assistant is used I receive feedback during class time	33%	40%	27%	0%	0%	0%
Send/Receive code is easier and faster than with other tools	71%	15%	11%	1%	2%	0%
Seeing other people's code helps me learn better	47%	29%	18%	3%	0%	3%

In both surveys, we had an open-ended question to let students express their opinion. We also made a focus group at the end of the term. The main comments received were as follows:

- Most of the students agreed with the idea that the class sessions where the tool was used were more active and efficient because the professor could review more solutions from her desk instead of going to each student's desk and because of the fast and easy way of sharing code.
- Students usually received feedback quickly when the tool was used.
- Some people commented that watching the professor explaining the errors in each solution helped them to learn better.
- Some students mentioned that reviewing other students' solutions helped them to see other ways of solving the same problem; thus, it helped them to learn better.

We also received some recommendations:

- The main recommendation students made was that the tool should have a way for the professor to type her feedback to each submission.
- Another recommendation was to have an easy way to know if an answer was right or wrong since students argued that when they reviewed the submissions of an activity, they had already forgotten which were correct and which were incorrect.

B. The Authors' opinion

The professor's comments are as follows:

- All the students have the opportunity to participate sending their solutions.
- More students get feedback of their work in each activity in the class session.
- The class is more agile because she can review the student's submissions in the moment they are sent and because of the fast and easy way of sharing code.
- The professor can have a good sense about how the group understands the topic. If she sees the same mistake several times, she can explain the concept again to the whole group so more people can understand it.

- Students who do not like to participate speaking in class sessions have the opportunity to receive feedback of their submissions.
- Students have the opportunity to correct their errors and send the solution again; thus, they can receive feedback several times during the same session in regards to the same problem.

VI. CONCLUSIONS

Using InClass Assistant has represented a positive experience for the students and the professor using it.

- In general, students have acknowledged that the class sessions are more active and flow more smoothly.
- Students have agreed that they usually receive feedback faster when the tool is used.
- Most of the students have said that they participate more in class when InClass Assistant is used.
- Several students think that watching other people's solutions helped them to learn better.
- Most of the students preferred a class in which InClass Assistant was used in comparison to sessions in which it was not used.
- The professor can give feedback to more people during the class session using the tool.
- Sharing people's solutions with the group helps students learn different ways to solve the same exercise.
- The tool can help people to realize that most people make mistakes in solving programming exercises when they are learning, so some students can improve their self-esteem based on this observation.
- For the professor, it is faster and easier to share code this way, compared with having to save the file with a representative name, and then upload it on the official platform.

FUTURE WORK

We are now working in the second version of InClass Assistant. Its main improvements are as follows:

- To be able to mark each submission as correct or incorrect.
- To be able to type a feedback comment to each submission. This will allow giving feedback even when the class time has finished.
- When a student sends a solution, the tool will ask him if he wants to make a new submission or if he wants to substitute the last one, avoiding this way that the same student has several submissions of the same exercise, when only small changes were made.

REFERENCES

- [1] Ambrosio, A. P., & Martins, S. W. 2011. What Matters Most When Teaching CS1? ITiCSE 2011 (p. 385). Darmstadt, Germany: ACM.
- [2] Guzdial, M. 2011. From science to engineering. In *Communications of the ACM*, 54 (2), pp. 37-39.
- [3] Briggs, T. (2005). Techniques for active learning in CS courses. *Journal of Computing Sciences in Colleges*, 21 (2), 156-165.
- [4] Gao, J., & Hargis, J. (2010). Promoting Technology- assisted Active Learning in Computer Science Education. *The Journal of Effective Teaching*, 10 (2), 81-93.
- [5] Martínez-Treviño, Y. (2013) What makes the difference, the active learning activities or the technology in the classroom. FECS 2013 (pp. 170-175). Las Vegas: CSREA Press.
- [6] Wilkerson, M., Griswold, W., & Simon, B. (2005). Ubiquitous Presenter: Increasing Student Access and Control in a Digital Lecturing Environment. SIGCSE 2005 (pp. 116-120). St. Luis, MO, USA: ACM.
- [7] Martínez-Treviño, Y., & Alarcón-Opazo, H. (2008). Uso de Tecnología para mejorar la comunicación durante la clase., CHIGE 2008. Monterrey, N.L. México: ITESM.
- [8] Shannon at al. (2016) PeerPresenter: A Web-Based System for In-Class Peer Feedback during Students Presentation. Proceedings of DIS, 2016. Pp 447-458. Brisbane, Australia.
- [9] Jung, M. (2016) Mark-On: Encouraging Student Questions in Class. Proceedings of the HCI. pp 79-86. Korea.
- [10] Scott, T. A. (2006). Computer Science Laboratory Improvement. CCSC: Rocky Mountain Conference (pp. 144-150). CCSC.
- [11] Rick, J. et al. (2002). Collaborative Learning at Low Cost: CoWeb Use in English Composition. Proceedings of CSCL, (pp. 435-442).