Online Mob Programming: Bridging the 21st Century Workplace and the Classroom

Sreecharan Sankaranarayanan, Xu Wang, Cameron Dashti, Marshall An, Clarence Ngoh, Michael Hilton, Majd Sakr, and Carolyn P. Rosé
sreechas@cs.cmu.edu, xuwang@cs.cmu.edu, cdashti1@cs.cmu.edu, haokanga@cs.cmu.edu, pcn@cs.cmu.edu, mhilton@cs.cmu.edu, msakr@cs.cmu.edu, cprose@cs.cmu.edu
School of Computer Science, Carnegie Mellon University

Abstract: We investigate how an industry-standard collaborative software development paradigm can be adapted for collaborative project-based learning in the classroom. The synchronous face-to-face collaboration paradigm, called Mob Programming, inspires Online Mob Programming (OMP), which structures groups of 3-6 students collaborating online in a rotating set of 3 roles supported by a conversational computer agent who takes on a 4th role. Results comparing OMP scaffolding with self-organization in a university computer science course shows OMP scaffolds encourage role-taking, division of labor, and conceptual reflection during work without a significant drop in group product quality.

Introduction

With a goal of integrating learning opportunities into collaborative work, we start in the software engineering context by adapting an industry inspired paradigm for collaborative software development called Mob Programming (Zuill and Meadows, 2016) into a collaborative learning paradigm for the classroom called Online Mob Programming (OMP). By focusing on a paradigm already popular in the industry, we aim for two distinct benefits: first, allowing students to engage in an industry relevant group practice in order to prepare them for work in industry; and second, developing an empirical foundation for structuring work practices in ways that might facilitate injecting collaborative learning opportunities into the workplace of the future. Evidenced by the rise of standards such as 21st Century Skills (Burrus et al., 2013), providing workplace-relevant learning opportunities for students is an increasing concern even as the rate at which the job market is changing is rapidly increasing.

Online Mob Programming is adapted from the industrial practice of Mob Programming, where participants rotate through the following roles – ***Driver***: A single participant who converts high-level instructions from the Navigator into code**;** ***Navigator***: A single participant who makes decisions from discussing with the Mob and communicates that to the Driver to be implemented into code**;** ***Mob***: A participant or group of participants who consider and deliberate between multiple alternative implementations ultimately informing the decision of the Navigator**;** ***Facilitator***: A single participant who observes and intervenes when necessary, such as to indicate when roles are to switch and to keep the activity progressing. The rotation of mob roles affords participants the opportunity to experience how group processes change when leadership changes within a group. Each participant will experience all the roles throughout a single mob programming session.

We address the concern that students in project courses fall into a pattern of emphasizing productivity over learning, which leads to divide-and-conquer strategies rather than shared cognition and conceptual reflection during work. In our work, the role of the facilitator is meant to guide role taking in order to mitigate these problems. That role is performed by an intelligent conversational agent implemented using the open source Bazaar framework (Adamson et al., 2014). Our study investigates the extent to which assigning students to OMP roles and periodic rotation of the role assignments reduces the adoption of suboptimal strategies. Through the interaction afforded by the paradigm, students are exposed to different perspectives in solving problems, building solutions, experimenting, debugging and writing readable code. They are also forced to externalize their thinking, which provides the opportunity for knowledge gaps to be revealed and addressed. Finally, they have the opportunity to observe knowledge and expertise in action as they observe their team-mates.

Experiment and results

The OMP framework discourages the allocation of tasks purely on the basis of prior expertise, thus allowing students to not only contribute in roles they are already good at, but also learn from their teammates to contribute in roles when they are not. We can hypothesize therefore, that (1) ***the OMP scaffold, if effective, will produce distinct collaborative behaviors associated with each role in mob programming.*** If we also believe that students would default to optimizing for productivity in the absence of such as a scaffold, we can hypothesize that (2) ***these distinct collaborative behaviors will not be adopted in self-organized groups, which might result in student behavior looking far more consistent throughout the activity.*** By enforcing the OMP scaffold for collaboration however, we run the risk that productivity may be harmed because students less expert at each subtask may get in the way. Furthermore, the cognitive load from role-switching might reduce productivity on the task and putting students in roles they are not familiar with could increase discomfort and therefore negatively affect their perception of the task. We hypothesize therefore, that (3) ***students from the OMP scaffold groups might feel more negatively about their experience compared to students from the self-organized groups and might perform worse on their project.*** In order to test our hypotheses, we experimentally contrast the mob programming scaffold against student self-organization in a between subjects design embedded within a *completely online* software development course. A total of 120 students took the course organizing themselves into teams of 3 for a semester long course project. In the first week of the course, they were grouped randomly into teams based only on their time availability to participate in an OMP training session. The experimental manipulation took place two weeks after the OMP training session when students had acquired the prerequisites necessary to complete the programming task. We collected logs of code contributions and chat logs from the team programming activity, Grades on individual assignments and the team project prior to and post the team programming exercise help control for differences in prior knowledge among students assigned to either condition, and a Post team programming exercise survey asking about prior familiarity with teammates, how this activity helped discover teammates', how they chose to structure the activity, how their experience with the OMP training session helped structure this activity, how effective they felt their organization was, both in the training session and in this activity, and their experience with the Cloud9 interface.

Hypothesis 1: The OMP scaffold, if effective, will produce distinct collaborative behaviors associated with each role in mob programming. A goal of OMP is to orchestrate rotation of team members through a set of three distinct but interdependent roles. We therefore expect to see distinctive behavior patterns within the collected data streams associated with the roles. As both a manipulation check and a lens to elucidate the effect of the manipulation on collaborative processes, we conducted a quantitative discourse analysis in the form of a factor analysis over the text. We expected and indeed saw characteristic patterns between roles to be more distinctive in the OMP condition. Hypothesis 1 was thus supported.

Hypothesis 2: The distinct collaborative behaviors associated with roles in the OMP condition will not be observed in self-organized groups, which might result in student behavior looking far more consistent throughout the activity. We considered that it is possible that within self-organized groups that members took up roles despite not having been assigned. If this was the case, we might not see those distinctively in the analysis above since turns from all team members are taken together in the self-organized condition and thus, we would only be able to see average behavior across roles (if any). In order to test this hypothesis therefore, we had to adopt a different methodology - cluster analysis. We clustered turns using k-means clustering in order to identify cross-cutting factor profiles. One cluster contained only turns from the 3 supported roles. None of the clusters were distinguishing for the unsupported condition. Hypothesis 2 is thus supported.

Hypothesis 3: Students from the OMP scaffolded groups might experience more discomfort as compared to students from the self-organized groups if the scaffolds are effective in countering the natural tendencies of students to gravitate towards what they are experienced doing. They may also produce lower quality work. In the post-programming exercise survey, we asked students about their experience with the Cloud 9 environment and how prepared they were for future group projects. The evidence that students in the Mob programming condition were less comfortable was marginal. Importantly however, a linear regression model built using the mob session grade as the outcome variable, the condition as the main factor while controlling for students' prior grades and familiarity with their team members found no significant differences between the two conditions suggesting that while students provided with the scaffold experienced marginally significantly more discomfort, this did not manifest itself in actual performance differences. Hypothesis 3 is thus partly supported.

References

Burrus, J., Jackson, T., Xi, N., & Steinberg, J. (2013). Identifying the most important 21st century workforce competencies: An analysis of the Occupational Information Network (O* NET). *ETS Research Report Series*, *2013*(2), i-55.

Zuill, W., & Meadows, K. (2016). Mob Programming: A whole Team Approach. In *Agile 2014 Conference, Orlando, Florida*.

Adamson, D., Dyke, G., Jang, H., & Rosé, C. P. (2014). Towards an agile approach to adapting dynamic collaboration support to student needs. *International Journal of Artificial Intelligence in Education*, *24*(1), 92-124.

Buchan, J., & Pearl, M. (2018, June). Leveraging the Mob Mentality: An Experience Report on Mob

Acknowledgements

This work was funded in part by NSF grants IIS 1822831 and IIS 1546393, and funding from Google.