Creating Opportunities for Transactive Exchange for Learning in Performance-Oriented Team Projects

Sreecharan Sankaranarayanan, Siddharth Reddy Kandimalla, Sahil Hasan, Haokang An, Christopher Bogart, R. Charles Murray, Michael Hilton, Majd Sakr, and Carolyn Rosé
sreechas@andrew.cmu.edu, skandima@andrew.cmu.edu, sahilh@andrew.cmu.edu,
haokanga@andrew.cmu.edu, cbogart@andrew.cmu.edu, rcmurray@andrew.cmu.edu,
mhilton@andrew.cmu.edu, msakr@andrew.cmu.edu, cprose@andrew.cmu.edu
Carnegie Mellon University

Abstract: Pursuing productivity, students often adopt a divide-and-conquer strategy that undercuts collaborative learning opportunities. In this study, we introduce a task structuring and role scaffolding paradigm to create opportunities for transactive exchange in such performance-oriented tasks and experimentally compare two prompting strategies -- one designed to create a focused discussion and another to intensify transactivity -- while controlling for time on task. We find significant learning gains of each strategy when used separately, but not in tandem.

Introduction and intervention design

Collaborative tasks are often designed so that task success requires learners to work together in an interdependent fashion and consequently, achieving success is a positive indicator of collaborative knowledge building. Tasks not intentionally designed this way however, can often be completed more efficiently by having individuals focus on parts of the task that align best with their existing expertise (a divide and conquer strategy). In doing so, individuals lose the opportunity to participate in collaborative learning. However, in many real-world tasks, such as software development, it may be valuable to inject opportunities for collaborative learning during productivity-focused tasks such as programming, resulting in a valuable tool for professional training during work. Consequently, we introduce a task structuring and role scaffolding paradigm, built based on the principles of the Script Theory of Guidance (Fischer et al., 2013), to set up opportunities for transactive exchange and then investigate two prompts – task-specific *Information prompts*, which seek to bring a discussion focus to the conversation; and task-agnostic *Transactivity prompts*, which seek to intensify transactivity.

As a step towards eventual adoption in the industry, this study is conducted in an online project-based programming course on Cloud Computing offered to graduate and upper-level undergraduate students at Carnegie Mellon University and its worldwide campuses. During the activity, students are placed into groups of 4 and assigned to four complementary roles based on the industry paradigm of Mob Programming (Wilson, 2015; Zuill & Meadows, 2016; Sankaranarayanan et al., 2019) – the *Navigator* is responsible for collecting ideas from the group and deciding on one which is implemented by the *Driver* assisted by the *Researcher*, who looks up resources such as a provided task primer and other web support material, as necessary, and the *Project Manager*, who is responsible for ensuring that the other team members are complying and adequately performing their roles.

The activity, lasting a total of 80 minutes, is divided into 5 tasks, each corresponding to a learning objective. Students first enter the programming phase where they assume their roles and work on the programming task. The completion of that task leads to a discussion phase where the group is supported by the conversational agent in a discussion based on the task. This task structuring can be considered a macroscript, i.e., an external collaboration script for sequencing an activity into learning phases (Dillenbourg & Hong, 2008). The agent supports the discussion using discourse-level prompts which can be considered microscripts i.e., explicit scaffolding for collaborative behavior within each learning phase. At the end of the discussion phase, the Project Manager reports on the team's level of compliance and performance on the task before the group moves on the next task and the roles rotate. This role assignment and rotation scaffold can also be considered a microscript.

Since roles remain the same within each task, but vary systematically across tasks, we can separately measure the learning that occurred during each task based on pre- and post-tests that students take before and after the activity containing 2 questions corresponding to each of the learning objectives (10 questions total). Student performance on a subsequent individual project associated with the task serves as a delayed post-test.

Based on the theoretical foundations of task design, the combination of the task structuring and role scaffolding should bring the discussion into focus and offer the opportunity for learning. However, there are many reasons why the reflective discussion may not occur in a way that is conducive to learning.

First, students may choose not to reflect on ideas that are meant for learning from the task. To address this possibility, the information prompts encourage the prompted student to make explicit their mental model about the relevance of the task to the learning objective (*Ex: "Ok, @Researcher, what is the advantage of writing*

OS-aware code like you have done here."). In doing so, their articulation serves as an opportunity for others to examine their own mental models against the one presented and discuss to reconcile any emergent conflicts. Because the prompts are designed for reflection on specific knowledge, they also serve as a form of direct instruction. Direct instruction is an efficient form of instruction and can lead to quicker knowledge convergence. It might not, however, foster subject matter expertise that generalizes beyond the specific knowledge provided.

Thus, a second concern is that students may discuss the target knowledge too briefly to elicit the deep reflection required for learning. To address this concern, the *transactivity prompts* provides time for an explanation attempt by one student and then prompts another student in the team to build on that attempt (*Ex: "@Researcher, do you agree with this approach? @Project Manager, what do you think? Think about how it fits into the larger implementation.")*. The transactivity prompts, in contrast to the information prompts, do not require a task-specific design but might result in additional time for reflection which may adversely affect task efficiency.

Method

Our hypotheses were tested with a 2x2 factorial design where the first factor was the presence or absence of *information prompts*, and the second factor was the presence or absence of *transactivity prompts*. Teams were randomly placed in four experimental conditions: 7 teams in the control, 6 in the condition where transactivity prompts were presented, 9 where both were presented, and 5 where only information prompts were presented. The control condition tested if the task structuring macroscript and the role scaffolding alone can bring enough discussion focus to impact learning. The two subsequent experimental conditions tested if the transactivity prompts when provided on their own or in tandem with information prompts would intensify this learning effect. The fourth condition tested only the information prompts. Testing the effects of transactivity prompts both with and without information prompts enables us to test whether the task structure and role scaffolding were sufficient to set the stage for transactive exchange. An interaction effect where transactivity prompts only have an effect in conjunction with the information prompts would suggest a failing in the task structure and role scaffolding.

Results and discussion

Significant pre- to post-test gains on tasks 2 and 3 out of the five were seen. Further, both prompts resulted in significant improvements in learning (while controlling for time on task) when presented alone but not when in tandem. We can conclude that the information prompts were superfluous given the role of the task structuring and role scaffolding in producing enough of a discussion focus. Further, presenting both prompts in tandem was counterproductive suggesting possible overscripting (Dillenbourg, 2002) potentially disrupting the "natural interactions" that were occurring with the transactivity prompts alone. Neither type had a significant positive effect on delayed post-test performance but trended positively, consistent with prior work (Noroozi et al., 2013). In summary, the combination of macroscript structuring and microscript prompts and role scaffolding presented here show promise as a possible means of introducing learning opportunities into productivity-focused projects.

References

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design.

Dillenbourg, P., & Hong, F. (2008). The mechanics of CSCL macro scripts. International Journal of Computer-Supported Collaborative Learning, 3(1), 5-23.

Fischer, F., Kollar, I., Stegmann, K., & Wecker, C. (2013). Toward a script theory of guidance in computer-supported collaborative learning. *Educational psychologist*, *48*(1), 56-66.

Noroozi, O., Weinberger, A., Biemans, H. J., Mulder, M., & Chizari, M. (2013). Facilitating argumentative knowledge construction through a transactive discussion script in CSCL. *Computers & Education*, *61*, 59-76.

Sankaranarayanan, S., Wang, X., Dashti, C., An, M., Ngoh, C., Hilton, M., ... & Rosé, C. (2019, June). An Intelligent-Agent Facilitated Scaffold for Fostering Reflection in a Team-Based Project Course. In *International Conference on Artificial Intelligence in Education* (pp. 252-256). Springer, Cham.

Wilson, A. (2015, May). Mob programming-what works, what doesn't. In *International Conference on Agile Software Development* (pp. 319-325). Springer, Cham.

Zuill, W., & Meadows, K. (2016). Mob programming: A whole team approach. In *Agile 2014 Conference, Orlando, Florida* (Vol. 3).

Acknowledgments

This work was funded in part by NSF grants IIS 1822831, IIS 1917955 and funding from Microsoft.