

Adapting Between Parsons Problems and Coding Tasks

Nadia L. Goralski
nadia.goralski@ontariotechu.net
Ontario Tech University
Oshawa, ON, Canada

Jeremy S. Bradbury
jeremy.bradbury@ontariotechu.ca
Ontario Tech University
Oshawa, ON, Canada

ABSTRACT

Previous research has shown that Parsons problems are an effective scaffolding activity for coding. Recently the development of Adaptive Parsons problems has provided more flexible scaffolding for students learning to code. However, there is still a gap between Parsons problems and coding tasks which can both challenge and frustrate novices. As such, we have developed an adaptive learning tool which looks to bridge this gap and support transitioning directly between Parsons problems and code writing.

KEYWORDS

Adaptive Learning, Computer Science Education, Parsons problems, Coding

ACM Reference Format:

Nadia L. Goralski and Jeremy S. Bradbury. 2023. Adapting Between Parsons Problems and Coding Tasks. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023)*, March 15–18, 2023, Toronto, ON, Canada. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/3545947.3576235>

1 INTRODUCTION

Parsons problems is a scaffolding technique for students which involves rearranging blocks of code to form a correct solution [6]. Prior research has shown it to be effective by allowing students to focus on learning how code works without the frustration of encountering syntactic errors [3].

Variations of Parsons problems have emerged that introduce new learning challenges as well as improve code comprehension. For example, adding in incorrect blocks called distractors [2] or in Faded Parsons problems requiring blank variable names and values to be filled in by the learner [7]. Another variation that provides more flexibility is adaptive Parsons Problems [3]. Adaptive Parsons allow the learner to adjust aspects of the problem such as the size of code blocks being arranged and the number of distractors used.

While Parsons problems are a scaffolding technique for learning coding, they are often viewed as a separate learning activity from coding. The gap between Parsons Problems and coding tasks leads to several questions including: (1) How should students transition from a Parsons problem to a coding task? (2) If a student struggles with a coding task is there value in transitioning back to a Parsons problem?

Permission to make digital or hard copies of part or all 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. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGCSE '23, March 15–18, 2023, Toronto, ON, Canada.

© 2023 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9433-8/23/03.

<https://doi.org/10.1145/3545947.3576235>

Our tool extends adaptive Parsons problems to include adapting directly between a challenging Parsons problem task and a coding task. We next plan to assess the learning benefits of our tool through a user study. We are also interested in assessing whether the ability to adapt directly from a Parsons problem to a coding task and back reduces the learner frustration reported in earlier studies [4, 5].

2 APPROACH

Our new adaptive Parsons problems tool includes the adaptation strategies developed in previous work [3] and extending it to include the ability to adapt directly to and from a coding task. This allows students to work at a self-selected level of difficulty from a Parsons problem with no distractors all the way up to writing code. Specifically, students can move between five levels:

- (1) two-line blocks of code with no distractors
- (2) two-line blocks of code with a low number (e.g., 1-2) of distractors
- (3) one-line blocks of code with a low number of distractors
- (4) one-line blocks of code with a moderate number (e.g., 3-4) of distractors
- (5) writing code in an online editor

We adapt from a Parsons problem to a code task by moving any solution code in the Parsons problem to the editor. Adapting from code back to a Parsons problem is considerably more challenging due to the free-form nature of coding where a learner may use different variable names and coding choices than those included in the solution code as well as make different errors than those included as distractors. To support this transition we use clone detection [1] to map learner written code to the Parsons problem solution code. We also incorporate learner generated mistakes as distractors and utilize learner created variable names in the Parsons problem task to enhance comprehension.

REFERENCES

- [1] James R Cordy and Chanchal K Roy. 2011. The NiCad clone detector. In *Proc. of the 19th Int. Conf. on Program Comprehension (ICPC 2011)*. IEEE, 219–220.
- [2] Yuemeng Du, Andrew Luxton-Reilly, and Paul Denny. 2020. A review of research on Parsons problems. In *Proc. of the 22nd Australasian Computing Education Conference (ACE'20)*. 195–202.
- [3] Barbara Ericson, Austin McCall, and Kathryn Cunningham. 2019. Investigating the affect and effect of adaptive Parsons problems. In *Proc. of the 19th Koli Calling Int. Conf. on Computing Education Research (Koli Calling '19)*. 1–10.
- [4] Carl C Haynes and Barbara J Ericson. 2021. Problem-solving efficiency and cognitive load for adaptive Parsons problems vs. writing the equivalent code. In *Proc. of the Conf. on Human Factors in Computing Systems (CHI 2021)*. 1–15.
- [5] Juha Helminen, Petri Ihanntola, Ville Karavirta, and Satu Alaoutinen. 2013. How do students solve Parsons programming problems? – execution-based vs. line-based feedback. In *2013 Learning and Teaching in Computing and Engineering*. 55–61.
- [6] Dale Parsons and Patricia Haden. 2006. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proc. of the 8th Australasian Conference on Computing Education-Volume 52 (ACE'06)*. 157–163.
- [7] Nathaniel Weinman, Armando Fox, and Marti A Hearst. 2021. Improving instruction of programming patterns with faded Parsons problems. In *Proc. of the Conf. on Human Factors in Computing Systems (CHI 2021)*. 1–4.