Planning a Multi-institutional and Multi-national Study of the Effectiveness of Parsons Problems

Barbara J. Ericson*
barbarer@umich.edu
University of Michigan
Ann Arbor, MI, USA

Paul Denny*
p.denny@auckland.ac.nz
The University of Auckland
Auckland, New Zealand

James Prather*
james.prather@acu.edu
Abilene Christian University
Abilene, Texas, USA

Rodrigo Duran
rodrigo.duran@ifms.edu.br
Fed. Institute of Mato Grosso do Sul
Nova Andradina, Brazil

Arto Hellas
arto.hellas@aalto.fi
Aalto University
Espoo, Finland

Juho Leinonen
juho.leinonen@helsinki.fi
University of Helsinki
Helsinki, Finland

Craig S. Miller
craig.miller@depaul.edu
DePaul University
Chicago, Illinois, USA

Briana Morrison
bbmorrison@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Janice L. Pearce
jan_pearce@berea.edu
Berea College
Berea, Kentucky, USA

Susan H. Rodger
rodger@cs.duke.edu
Duke University
Durham, North Carolina, USA

ABSTRACT

Programming is a complex task that requires the development of many skills including knowledge of syntax, problem decomposition, algorithm development, and debugging. Code-writing activities are commonly used to help students develop these skills, but the difficulty of writing code from a blank page can overwhelm many novices. Parsons problems offer a simpler alternative to writing code by providing scrambled code blocks that must be placed in the correct order to solve a problem. The extensive literature on Parsons problems documents numerous benefits to using them as both formative and summative assessments. These include more efficient learning, the possibility to dynamically adapt to learner needs, and more reliable grading. Despite these positive findings, further research is needed in order to draw broader inferences. Most work has been conducted at single institutions under unique conditions that are not easily replicated, and some prior studies have been inconclusive or had limitations that affected data validity. To address this, we propose a multi-institutional and multi-national study of the effectiveness of Parsons problems for novice programmers. We will focus on introductory programming courses (CS0/1/2) that use Java, Python, and C/C++ as these are the most common teaching languages. The working group will collaborate to refine the scope, methodology and research questions, and contribute to data collection and analysis.

CCS CONCEPTS

• **Social and professional topics** → *Computing education.*

KEYWORDS

Parsons Problems, Parsons Puzzles, Parson's Programming Puzzles, Parson's Problems, Parson's Puzzles, Code Puzzles

ACM Reference Format:

Barbara J. Ericson, Paul Denny, James Prather, Rodrigo Duran, Arto Hellas, Juho Leinonen, Craig S. Miller, Briana Morrison, Janice L. Pearce, and Susan H. Rodger. 2022. Planning a Multi-institutional and Multi-national Study of the Effectiveness of Parsons Problems. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol 2 (ITiCSE 2022), July 8–13, 2022, Dublin, Ireland.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3502717.3532172

1 BACKGROUND

Researchers have studied many variations of Parsons problems and have used a variety of names for them [1–3, 9, 11–14, 16, 17]. Parsons and Haden, who referred to them as *Parson's Programming Puzzles*, originally proposed them to maximize engagement, constrain the logic, permit common errors, model good code, and provide immediate feedback [16]. We use the term 'Parsons problems,' which was later coined by Denny, Luxton-Reilly, and Simon [4]. Despite increasing use of Parsons problems in computing classrooms, and a growing body of literature exploring their benefits, current findings are from single institutions. This suggests a need for replication work and large scale evaluations [5].

Compared to writing code from a blank page, Parsons problems are considered to be easier for novices as they greatly constrain the

*co-leader

ITiCSE 2022, July 8–13, 2022, Dublin, Ireland

ACM ISBN 978-1-4503-9200-6/22/07.
https://doi.org/10.1145/3502717.3532172

problem-solving space. A common variation that adds complexity is the use of distractor blocks, although studies have reported mixed findings. Harms et al. found that cognitive load was increased by the addition of distractors, leading to lower learning efficiency and rates of success when solving problems without distractors [10]. In contrast, Ericson found that distractors helped participants recognize errors in code and learn to write code [6]. From a theoretical standpoint, Margulieux *et al.* hypothesize that learners develop better conceptual knowledge when prompted to compare multiple conceptions, and highlight Parsons problems with distractors as an exemplar task [15]. This is just one example of an area that needs additional study with respect to Parsons problems.

2 GOALS

A key goal of the working group will be to conduct an extensive review of the literature that classifies prior research and presents a synthesis of empirical findings. We will explore the defining properties of Parsons problems, and present a framework that will be useful for capturing the many variations of Parsons problems that have been proposed. In this initial phase of the working group, relevant literature will be gathered from previous reviews and seminal work to form a corpus that will serve as a validation set for the proposed search. Alongside inclusion and exclusion criteria, key search terms will be defined and used to conduct the search on source databases. This extensive literature review will enable us to identify and publish a list of research questions that require further exploration. We will select a subset of these questions and design materials and protocols for their investigation which we will refine through small-scale pilot studies. Candidate research questions, drawn from recent prior research, include broad investigation of the effects of solving Parsons problems with and without distractors, and exploring the benefits and challenges of solving Parsons problems compared with writing the equivalent code. The experimental resources will be a key outcome of the working group, and will support follow-up research that can be conducted at a larger scale. Prior studies of the effectiveness of Parsons problems have been conducted at single institutions, so multi-national and multi-institutional studies will contribute important new knowledge.

When designing the experiments and resources, and conducting related pilot studies, we plan to leverage the open-source ebook platform, Runestone, which allows Parsons problems to be used in both formative and summative assessments [8]. Runestone is a robust platform for running experiments at scale, with tens of thousands of students and hundreds of institutions currently using computing ebooks that have been published on Runestone. These ebooks contain hundreds of Parsons problems. Runestone also supports novel variations of Parsons problems, such as adaptive Parsons problems, in which the difficulty of the problem is personalized based on the learner's performance [7]. Runestone also provides direct support for A/B testing, which facilitates experiments looking to establish causal effects.

3 CONCLUSION

Parsons problems have a long history in computing education practice, providing convenient scaffolding for students learning to program. Although various benefits to students and educators have been documented, many studies are inconclusive or conducted at a small scale suggesting a need for replication and further research to produce generalisable results. Our working group will lay the groundwork for such studies. Our extensive review of the literature will act as a useful starting point for those interested in learning about Parsons problems. The list of open research questions will guide computing education researchers exploring Parsons problems, and our experimental resources and protocols will facilitate this work. Our report will lay the foundation for future research on Parsons problems in computing education and influence how introductory programming is taught.

REFERENCES

[1] John Allen and Caitlin Kelleher. 2021. Quantifying Novice Behavior, Experience, and Mental Effort in Code Puzzle Pathways. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–6.

[2] ATM Golam Bari, Alessio Gaspar, R Paul Wiegand, Jennifer L Albert, Anthony Bucci, and Amruth N Kumar. 2019. EvoParsons: design, implementation and preliminary evaluation of evolutionary Parsons puzzle. *Genetic Programming and Evolvable Machines* 20, 2 (2019), 213–244.

[3] Nick Cheng and Brian Harrington. 2017. The Code Mangler: Evaluating Coding Ability Without Writing any Code. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 123–128.

[4] Paul Denny, Andrew Luxton-Reilly, and Beth Simon. 2008. Evaluating a new exam question: Parsons problems. In *Proceedings of the fourth international workshop on computing education research*. 113–124.

[5] Yuemeng Du, Andrew Luxton-Reilly, and Paul Denny. 2020. A Review of Research on Parsons Problems. In *Proceedings of the Twenty-Second Australasian Computing Education Conference* (Melbourne, VIC, Australia) *(ACE'20)*. Association for Computing Machinery, New York, NY, USA, 195–202. https://doi.org/10.1145/3373165.3373187

[6] Barbara Ericson, Austin McCall, and Kathryn Cunningham. 2019. Investigating the Affect and Effect of Adaptive Parsons Problems. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*. 1–10.

[7] Barbara J Ericson, James D Foley, and Jochen Rick. 2018. Evaluating the efficiency and effectiveness of adaptive parsons problems. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. 60–68.

[8] Barbara J Ericson and Bradley N Miller. 2020. Runestone: A Platform for Free, Online, and Interactive Ebooks. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 1012–1018.

[9] Rita Garcia. 2021. Evaluating Parsons Problems as a Design-Based Intervention. In *2021 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9.

[10] Kyle James Harms, Jason Chen, and Caitlin L Kelleher. 2016. Distractors in Parsons problems decrease learning efficiency for young novice programmers. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*. 241–250.

[11] Kyle J Harms, Noah Rowlett, and Caitlin Kelleher. 2015. Enabling independent learning of programming concepts through programming completion puzzles. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 271–279.

[12] Petri Ihantola and Ville Karavirta. 2010. Open source widget for parson's puzzles. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*. 302–302.

[13] Petri Ihantola and Ville Karavirta. 2011. Two-dimensional parson's puzzles: The concept, tools, and first observations. *Journal of Information Technology Education* 10, 2 (2011), 119–132.

[14] Amruth N Kumar. 2019. Helping Students Solve Parsons Puzzles Better. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. 65–70.

[15] Lauren Margulieux, Paul Denny, Kathryn Cunningham, Michael Deutsch, and Benjamin R. Shapiro. 2021. When Wrong is Right: The Instructional Power of Multiple Conceptions. In *Proceedings of the 17th ACM Conference on International Computing Education Research* (Virtual Event, USA) *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 184–197. https://doi.org/10.1145/3446871.3469750

[16] Dale Parsons and Patricia Haden. 2006. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*. 157–163.

[17] Nathaniel Weinman, Armando Fox, and Marti A. Hearst. 2021. Improving Instruction of Programming Patterns with Faded Parsons Problems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 53, 4 pages. https://doi.org/10.1145/3411764.3445228