

How Novice Programmers Interact with Programming Environments

Keynote

2019 International Workshop on Computer Science Education (IWCSE 2019)

Changsha, Hunan, China

Brett A Becker

University College Dublin & Beijing-Dublin International College

Dublin, Ireland & Beijing, China

brett.becker@ucd.ie

www.brettbecker.com

ABSTRACT

Teaching and learning programming is fraught with challenges, many unique to computing education. While learning to program, students are not only required to learn the fundamental concepts of programming itself, but also to grasp the syntax and grammar of a programming language. Typically this is achieved through one of many programming environments. Common to all of these environments is that they provide feedback in the form of error messages. However, how students interact with these messages is an under-studied area. Exactly how novices interact with these environments and their error messages varies widely, and there seems to be little consistency between various tools, languages, and environments. The result is a myriad of different modes of interaction: text, pop-ups, notifications, syntax highlighting, auto-completing code, background compilation, and more. This keynote aims to open the question of why and how these various means of interaction differ, what the possible effects of this are, and what should be studied in order to improve how students learn to program.

REFERENCES

- [1] Amjad Altadmri, Michael Kölling, and Neil C. C. Brown. 2016. The Cost of Syntax and How to Avoid It: Text versus Frame-Based Editing. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, 748–753. <https://doi.org/10.1109/COMPSAC.2016.204>
- [2] T Barik. 2018. *Error Messages as Rational Reconstructions*. Ph.D. Dissertation. <https://repository.lib.ncsu.edu/bitstream/handle/1840.20/35439/etd.pdf?sequence=1>
- [3] B.A. Becker, C. Murray, T. Tao, C. Song, R. McCartney, and K. Sanders. 2018. Fix the first, ignore the rest: Dealing with multiple compiler error messages. In *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, Vol. 2018-Janua. <https://doi.org/10.1145/3159450.3159453>
- [4] Douglas Blank, Jennifer S. Kay, James B. Marshall, Keith O'Hara, and Mark Russo. 2012. Calico: A Multi-Programming-Language, Multi-Context Framework Designed for Computer Science Education. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education - SIGCSE '12*. ACM Press, New York, New York, USA, 63. <https://doi.org/10.1145/2157136.2157158>
- [5] Dennis Bouvier, Mark Zarb, Ellie Lovellette, John Matta, Bedour Alshaiqy, Brett A. Becker, Michelle Craig, Jana Jackova, Robert McCartney, and Kate Sanders. 2016. Novice Programmers and the Problem Description Effect. In *Proceedings of the 2016 ITiCSE Working Group Reports on - ITiCSE '16*. ACM Press, New York, New York, USA, 103–118. <https://doi.org/10.1145/3024906.3024912>
- [6] Carla De Lira. 2017. Improving the Learning Experiences of First-Year Computer Science Students with Empathetic IDEs. In *Proceedings of the 2017 ACM Conference on International Computing Education Research - ICER '17*. ACM Press, New York, New York, USA, 293–294. <https://doi.org/10.1145/3105726.3105742>
- [7] Gregory Dyke. 2011. Which aspects of novice programmers' usage of an IDE predict learning outcomes. In *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11*. ACM Press, New York, New York, USA, 505. <https://doi.org/10.1145/1953163.1953309>
- [8] Christoph Hannebauer, Marc Hesenius, and Volker Gruhn. 2018. Does syntax highlighting help programming novices? *Empirical Software Engineering* 23, 5 (oct 2018), 2795–2828. <https://doi.org/10.1007/s10664-017-9579-0>
- [9] Jason T. Jacques and Per Ola Kristensson. 2015. Understanding the effects of code presentation. In *Proceedings of the 6th Workshop on Evaluation and Usability of Programming Languages and Tools - PLATEAU 2015*, Vol. 28. ACM Press, New York, New York, USA, 27–30. <https://doi.org/10.1145/2846680.2846685>
- [10] Matthew C. Jadud. 2006. Methods and tools for exploring novice compilation behaviour. In *Proceedings of the 2006 international workshop on Computing education research - ICER '06*. ACM Press, New York, New York, USA, 73. <https://doi.org/10.1145/1151588.1151600>
- [11] Hyeonsu Kang and Philip J. Guo. 2017. Omnicode: A Novice-Oriented Live Programming Environment with Always-On Run-Time Value Visualizations. *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST '17* (2017), 737–745. <https://doi.org/10.1145/3126594.3126632>
- [12] Michael Kölling, Neil C. C. Brown, Hamza Hamza, and Davin McCall. 2019. Stride in BlueJ - Computing for All in an Educational IDE. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19*. ACM Press, New York, New York, USA, 63–69. <https://doi.org/10.1145/3287324.3287462>
- [13] Michael Kölling, Bruce Quig, Andrew Patterson, and John Rosenberg. 2003. The BlueJ System and its Pedagogy. *Computer Science Education* 13, 4 (dec 2003), 249–268. <https://doi.org/10.1076/csed.13.4.249.17496>
- [14] Benjamin Lafreniere, Parmit K Chilana, Adam Fourney, and Michael A Terry. 2015. These Aren't the Commands You're Looking For. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology - UIST '15*. ACM Press, New York, New York, USA, 619–628. <https://doi.org/10.1145/2807442.2807482>
- [15] Marie-Hélène Nienaltowski, Michela Pedroni, and Bertrand Meyer. 2008. Compiler error messages: What can help novices? *Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE 08)* 40, 1 (feb 2008), 168. <https://doi.org/10.1145/1352322.1352192>
- [16] James Prather, Raymond Pettit, Kayla Holcomb McMurry, Alani Peters, John Homer, Nevan Simone, and Maxine Cohen. 2017. On Novices' Interaction with Compiler Error Messages. In *Proceedings of the 2017 ACM Conference on International Computing Education Research - ICER '17*. ACM Press, New York, New York, USA, 74–82. <https://doi.org/10.1145/3105726.3106169>
- [17] Charles Reis and Robert Cartwright. 2005. Taming a professional IDE for the classroom. *ACM SIGCSE Bulletin* 36, 1 (2005), 156. <https://doi.org/10.1145/1028174.971357>
- [18] Mark Santolucito, William T Hallahan, and Ruzica Piskac. 2019. Live Programming By Example. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. ACM Press, New York, New York, USA, 1–4. <https://doi.org/10.1145/3290607.3313266>
- [19] Arto Vihavainen, Juha Helminen, and Petri Ihantola. 2014. How novices tackle their first lines of code in an IDE. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research - Koli Calling '14*. ACM Press, New York, New York, USA, 109–116. <https://doi.org/10.1145/2674683.2674692>
- [20] David Weintrop and Uri Wilensky. 2015. Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. In *Proceedings of the eleventh annual International Conference on International Computing Education Research - ICER '15*. ACM Press, New York, New York, USA, 101–110. <https://doi.org/10.1145/2787622.2787721>