Epistemic Programming as a Scientific Field: Building a Community of Practice and an Interaction-Based Framework

Sven Hüsing*
Paderborn University
Paderborn, Germany
sven.huesing@uni-paderborn.de

Line Have Musaeus*
Aarhus University
Aarhus, Denmark
lh@cs.au.dk

Michael E. Caspersen*
Aarhus University
Aarhus, Denmark
mec@cs.au.dk

Carsten Schulte*
Paderborn University
Paderborn, Germany
carsten.schulte@uni-paderborn.de

Erik Barendsen
Radboud University
Nijmegen, Netherlands
Open University
Nijmegen, Netherlands
erik.barendsen@ru.nl

Nataša Grgurina
University of Groningen
Groningen, Netherlands
The Netherlands Institute for Curriculum Development
Amersfoort, Netherlands
n.grgurina@rug.nl

Matthias Hauswirth
Università della Svizzera italiana
Lugano, Switzerland
matthias.hauswirth@usi.ch

Violetta Lonati
Università degli Studi di Milano
Milan, Italy
Laboratorio CINI Informatica e Scuola Rome
Rome, Italy
lonati@di.unimi.it

Murali Mani
University of Michigan-Flint
Flint, Michigan, United States
mmani@umich.edu

Mattia Monga
Università degli Studi di Milano
Milan, Italy
Laboratorio CINI Informatica e Scuola Rome
Rome, Italy
mattia.monga@unimi.it

Heidi Nobles
University of Virginia
Charlottesville, Virginia, United States
heidi.nobles@virginia.edu

Scott Reckinger
University of Illinois Chicago
Illinois, Chicago, United States
scotreck@uic.edu

Devin W. Silvia
Michigan State University
East Lansing, Michigan, United States
dsilvia@msu.edu

Sören Sparmann
Paderborn University
Paderborn, Germany
soeren.sparmann@uni-paderborn.de

Abstract

Programming is increasingly recognized not only as a technical skill but also as a means of exploration and insight generation in personal and societal contexts. In this regard, epistemic programming (EP) refers to programming for discovery, expression, and interdisciplinary learning. This working group (WG) aims to establish an international community, bringing together educators and researchers to develop a shared vision and contribute to EP as a scientific field. By analysing current practices and approaches, we will outline an interaction-based framework and a research agenda for further development and integration of EP into education.

*co-leader

ITiCSE 2026, Madrid, Spain

ACM ISBN 979-8-4007-2633-0/2026/07
https://doi.org/10.1145/3803401.3812051

CCS Concepts

• **Social and professional topics** → **K-12 education**; **Computing education**.

Keywords

programming education, epistemic programming, philosophy

ACM Reference Format:
Sven Hüsing, Line Have Musaeus, Michael E. Caspersen, Carsten Schulte, Erik Barendsen, Nataša Grgurina, Matthias Hauswirth, Violetta Lonati, Murali Mani, Mattia Monga, Heidi Nobles, Scott Reckinger, Devin W. Silvia, and Sören Sparmann. 2026. Epistemic Programming as a Scientific Field: Building a Community of Practice and an Interaction-Based Framework. In *Proceedings of the 31st ACM Conference on Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2026), July 10–15, 2026, Madrid, Spain.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3803401.3812051

1 Introduction & Background

Programming is a much debated competency in computer science education, also due to the existence of diverse "flavours" of programming practices [1]. In recent years, programming has become increasingly accessible to researchers from other disciplines as well as to laypersons, enabling its use in the pursuit of individual and societal interests [18, 19]. Within this context, a particularly prominent "flavour" of programming emphasizes its use as a means of *exploration and gaining insights*, which we refer to as Epistemic Programming (EP). The purpose of EP is the empowerment to computationally explore personal, societal, or disciplinary topics in a self-determined way [7]. In the context of EP, goals and expectations are typically not fixed in advance, but are influenced through the programming process itself, e.g. when a computed plot hints at a certain anomaly of the data. Thus, EP-processes are characterized through a conversational nature in an interaction between the programmer, the programming environment and the results [5], "extending" the programmer's mind [4, 10]. From this interaction the programmer iteratively can draw new insights by reflecting on the current state of the program and the results, enabling for new (epistemic) interests, and subsequent actions and adaptations of the program code, thereby fostering epistemic agency [12]. While this idea resonates with established concepts like Constructionism [13], Personal Dynamic Media [8] and Literate Programming [9] in sharing the view on programming as a personal activity for exploring, constructing and communicating insights (also see [6, 11, 15]), today, several approaches from diverse, interdisciplinary contexts focus on enabling (novice) programmers to use programming in this regard [2, 19] within an inquiry-based learning process [14] and a conversational approach with the program code [3, 5, 17]. Currently, however, there is no common perspective or framework for EP in education. The goal of this WG is therefore to build a community of practice and establish such a joint vision for EP.

2 Goals

The WG aims to establish an international and interdisciplinary community on EP to promote computing as a discipline for discovery and expression relevant for all. Therefore, we want to specify the notion of EP by (a) identifying characteristics both regarding EP as a discrete computing activity as well as in interdisciplinary contexts and by (b) deriving an initial interaction-based framework that allows for making EP-processes accessible for educators and learners as well as for analysing learners' processes to develop tailored teaching material. Following this, we seek to explore the state of affairs in EP education and outline a future research agenda together with and for the established community.

3 Methodology

First, the WG will develop analysis topics for analysing current EP-practices and processes. Through conducting (a) an artifact review (of e.g. educational standards, teaching material, and EP-products), (b) interviews with practitioners and educators, and (c) a literature review, the WG will compile findings regarding these analysis topics. Based on this, the WG will develop an interaction-based framework by specifying the hybrid-interaction-framework [16] for EP as a particular form of interaction. Building on this, potential research topics and a research agenda will be outlined together with the EP-community. For building this community, the WG will engage with an open network of associates, consisting of interested researchers and educators. The associates contribute through participating as interviewees and attending online-sessions on the current progress as well as on topics and literature in the context of EP.

References

[1] Ilias Bergstrom and Alan F. Blackwell. 2016. The practices of programming. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, Cambridge, United Kingdom, 190–198. doi:10.1109/VLHCC.2016.7739684

[2] Mary Beth Kery and Brad A. Myers. 2017. Exploring exploratory programming. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, Raleigh, NC, 25–29. doi:10/gqqmnp

[3] Joel Brandt, Philip J. Guo, Joel Lewenstein, and Scott R. Klemmer. 2008. Opportunistic programming: How rapid ideation and prototyping occur in practice. In *Proceedings of the 4th international workshop on End-user software engineering*. ACM, Leipzig Germany, 1–5. doi:10.1145/1370847.1370848

[4] Andy Clark and David Chalmers. 1998. The Extended Mind. *Analysis* 58, 1 (1998), 7–19. doi:10/gdm72m

[5] Kathryn Cunningham, Barbara J. Ericson, Rahul Agrawal Bejarano, and Mark Guzdial. 2021. Avoiding the Turing Tarpit: Learning Conversational Programming by Starting from Code's Purpose. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–15. doi:10.1145/3411764.3445571

[6] Mark Guzdial and August Evrard. 2023. Identifying the Computing Education Needs of Liberal Arts and Sciences Students (Discussion Paper). In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. ACM, Koli Finland, 1–7. doi:10.1145/3631802.3631805

[7] Sven Hüsing, Carsten Schulte, and Felix Winkelnkemper. 2023. Epistemic Programming. In *Computer Science Education: Perspectives on Teaching and Learning in School* (2 ed.), Sue Sentance, Erik Barendsen, Nicol R. Howard, and Carsten Schulte (Eds.). Bloomsbury Academic, London, 291–304. http://dx.doi.org/10.5040/9781350296947.ch-022

[8] Alan Kay and A. Goldberg. 1977. Personal Dynamic Media. *Computer* 10, 3 (March 1977), 31–41. doi:10/cdmm6c

[9] D. E. Knuth. 1984. Literate Programming. *Comput. J.* 27, 2 (Feb. 1984), 97–111. doi:10/c73jjw

[10] Line Have Musaeus and Peter Musaeus. 2021. Computing and Gestures in High School Biology Education. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. ACM, Virtual Event Germany, 533–539. doi:10.1145/3430665.3456307

[11] Palle Nowack and Michael E. Caspersen. 2014. Model-based thinking and practice: a top-down approach to computational thinking. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '14)*. Association for Computing Machinery, New York, NY, USA, 147–151. doi:10.1145/2674683.2674686

[12] Tor Ole B. Odden, Devin W. Silvia, and Anders Malthe-Sørenssen. 2023. Using computational essays to foster disciplinary epistemic agency in undergraduate science. *Journal of Research in Science Teaching* 60, 5 (2023), 937–977. doi:10.1002/tea.21821

[13] Seymour Papert and Idit Harel. 1991. *Situating Constructionism*. Ablex Publishing Corporation, Norwood, NJ, 1–12.

[14] Margus Pedaste, Mario Mäeots, Leo A. Siiman, Ton de Jong, Siswa A.N. van Riesen, Ellen T. Kamp, Constantinos C. Manoli, Zacharias C. Zacharia, and Eleftheria Tsourlidaki. 2015. Phases of inquiry-based learning: Definitions and the inquiry cycle. *Educational Research Review* 14 (2015), 47–61. doi:10.1016/j.edurev.2015.02.003

[15] Mitchel Resnick. 2013. Learn to code, code to learn. *EdSurge* 54 (2013). https://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf

[16] Carsten Schulte and Lea Budde. 2018. A Framework for Computing Education: Hybrid Interaction System: The need for a bigger picture in computing education. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*. ACM, Koli Finland, 1–10. doi:10.1145/3279720.3279733

[17] D.A. Schön. 1992. Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems* 5, 1 (March 1992), 3–14. doi:10.1016/0950-7051(92)90020-G

[18] Mike Tissenbaum, David Weintrop, Nathan Holbert, and Tamara Clegg. 2021. The case for alternative endpoints in computing education. *British Journal of Educational Technology* 52, 3 (May 2021), 1164–1177. doi:10.1111/bjet.13072

[19] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology* 25, 1 (Feb. 2016), 127–147. doi:10.1007/s10956-015-9581-5