Paper ID #17928

Process Oriented Guided Inquiry Learning in Computer Science: The CS-POGIL & IntroCS-POGIL Projects

Mr. Clifton L Kussmaul, Muhlenberg College

Clif Kussmaul is Associate Professor of Computer Science at Muhlenberg College. Previously, he was Visiting Fulbright-Nehru Scholar at the University of Kerala, Chief Technology Officer for Elegance Technologies, Inc., Senior Member of Technical Staff with NeST Technologies, and Assistant Professor of CS at Moravian College. He has a PhD in Computer Science from the University of California, Davis, master's degrees in CS and Electro-acoustic Music from Dartmouth College, and bachelor's degrees in Engineering and Music from Swarthmore College. His professional interests and activities include active and guided inquiry learning, software engineering, entrepreneurship, digital signal processing, cognitive neuroscience, and music.

Dr. Chris Mayfield, James Madison University

Chris Mayfield, Ph.D., is an Assistant Professor of Computer Science at James Madison University. His research focuses on CS education and professional development, including for K-12 schools. Over the past five years, he has taught introductory CS courses using POGIL and the flipped classroom. He is coauthor of the textbook Think Java: How to Think Like a Computer Scientist and the designer of JMU's CS 101 course.

Helen H Hu, Westminster College

Helen H. Hu received her Ph.D. in computer science from the University of Utah. She is a Professor of Computer Science at Westminster College and a member of the ACM. Her research interests include active learning pedagogies and broadening participation in computer science

Process Oriented Guided Inquiry Learning in Computer Science:
The CS-POGIL & IntroCS-POGIL Projects

Abstract

Process Oriented Guided Inquiry Learning (POGIL) is an evidence-based pedagogical approach in which student teams work on classroom activities designed to help them collaboratively construct understanding of key concepts, and at the same time to develop process skills including communication, critical thinking, problem solving, and teamwork. POGIL activities use learning cycles in which teams are given a model (e.g. a diagram, graph, table, or sample code) and then answer questions that guide them to explore the model, invent their own understanding of key concepts, and then apply that understanding in other contexts. The instructor is not a lecturer, but an active facilitator who observes student teams, interacts to address problems, and leads classroom discussion as needed. POGIL has been used across STEM disciplines (including chemistry, engineering, computer science, and mathematics), and research studies generally find that students have better learning outcomes.

The CS-POGIL project and the IntroCS-POGIL project both seek to expand the use and evidence of POGIL in Computer Science and related areas. The 2011 NSF TUES CS-POGIL project developed sample POGIL activities for topics in intermediate level CS courses, and helped to develop a POGIL community in CS through conference presentations, support for CS faculty to attend 3-day regional POGIL workshops, and online discussions. The 2017 NSF IUSE IntroCS-POGIL project is a larger-scale study of how faculty implement POGIL in introductory CS courses and the factors that affect faculty implementation and student outcomes.

Introduction

Process Oriented Guided Inquiry Learning (POGIL) is an evidence-based, student-centered pedagogy that develops both content knowledge and process skills (such as communication, critical thinking, problem solving, and teamwork). Students work in learning teams on specifically designed activities that guide them to construct key concepts. POGIL incorporates techniques that have been shown to retain students from underrepresented groups [e.g. 1, 2, 3]. POGIL has been used across STEM disciplines including engineering [4], mathematics [5], microbiology [6], and physiology [7]; in non-STEM fields such as environmental health [8], finance [9], and library science [10]. Research studies generally find that students in POGIL classes have better learning outcomes [e.g. 11, 12]. However, POGIL requires significant effort to develop or adapt materials and to implement effective classroom facilitation techniques. The POGIL Project (http://pogil.org) organizes professional development opportunities and fosters a community of practice across disciplines.

POGIL is especially appropriate for CS because it emphasizes process skills, and thus shifts student attention away from language syntax and towards conceptual understanding. For example, a POGIL activity might guide students through the problem-solving process of writing a loop by posing questions about its behavior, rather than emphasizing where the parentheses and braces need to go. By encouraging collaborative learning, POGIL improves outcomes for underrepresented groups, a problem more severe in CS than in most other STEM fields. After converting a CS1 course to POGIL, pass rates increased for female students but not males [13]. A POGIL CS Principles course increased student interest in taking additional CS courses [14]. In a software project course, POGIL activities helped students to understand the importance of communication in real software projects [15].

CS-POGIL Objectives & Activities

The 2011 NSF TUES CS-POGIL project had two broad objectives:

Objective I: Create effective, polished activities and outcome reports, so that POGIL can be adopted, adapted, and assessed by other faculty. This included two sets of activities for topics commonly taught in CS curricula, and focused on concepts rather than language-specific syntax. The first set focused on software engineering, including topics such as risk management, estimation, project planning, and the Unified Modeling Language (UML). The second set focused on data structures and algorithms, including algorithm analysis, searching, sorting, and linked data structures.

Objective II: Raise awareness of POGIL in computer science, software engineering, and related disciplines and foster a POGIL community in these disciplines. This included posters, presentations, and birds-of-a-feather sessions at conferences to raise awareness; and workshop and tutorial sessions to provide a deeper introduction. It also included support for CS faculty to attend longer professional development programs, including 3-day regional workshops organized by The POGIL Project.

CS-POGIL Outcomes

The CS-POGIL project developed, piloted, and revised over 50 activities (see http://cspogil.org), focusing on data structures and algorithms and software engineering but including other CS topics. In total, there are now over 200 POGIL activities for CS. CS-POGIL has fostered a growing POGIL community within CS, through conference sessions and workshops (20); support for 3-day workshops (25 faculty, including 4 at community colleges) and facilitator training (5 faculty); and a 60-member Google Group. Thus, CS-POGIL leveraged and extended the successful POGIL approach to a new area (CS), and explored tools and approaches that may interest the broader POGIL community.

The CS POGIL Google Group was surveyed on their use of POGIL. Of respondents who had tried POGIL, 92% believe it increases student engagement, and 84% believe it encourages deeper understanding. Q The majority also perceived improvements in students' effectiveness in working in small groups, their communication skills, and positive peer-to-peer relationships. The greatest obstacle to adopting POGIL was time (both preparation time and classroom time required). 71% believed that a closer community (e.g. face-to-face meetings with mentors or colleagues) would be somewhat, very, or extremely helpful to new POGIL adopters, and for regular users of POGIL this percentage increases to 87%. [16]

IntroCS-POGIL Objectives & Activities

The 2017 NSF IUSE IntroCS-POGIL project focuses on introductory courses because they: enroll the most students (across STEM majors) and have the biggest impact on retention; involve the most instructors and thus the largest potential community; and have the most diverse pool of existing POGIL activities. IntroCS-POGIL has four objectives:

Objective I: Refine and validate POGIL activities for IntroCS courses. Several sets of IntroCS POGIL activities (in Java and Python) have already been developed. These sets are being reviewed, refined, and expanded to provide broader coverage and more flexibility. Each activity has a 1-2 page instructor's guide with learning objectives and some preparation and facilitation notes. We will expand these guides with important structural components [17] to support instructors in varied settings, leading to broader propagation, as well as a useful model for POGIL authors in other disciplines.

Objective II: Provide ongoing support for IntroCS faculty who adopt POGIL. In response to feedback from the CS-POGIL community, we are augmenting the existing three-day POGIL training workshop with an extra day of CS-specific sessions. These will include: an overview of the available activities and instructor guides; discussions on how and when to implement POGIL in IntroCS courses; course planning time; and community building time. We will also pilot other resources for POGIL faculty, including: a catalog of POGIL practices as pedagogical patterns; short videos of POGIL faculty and practices; and new POGIL workshops for faculty.

Each year, we will recruit teams of new instructors to attend POGIL training workshops (see above) and implement POGIL in their IntroCS courses. Each instructor will create a portfolio of materials, and complete reflective teaching logs. Each mentor will observe and provide feedback on each member's classroom. All team members will join the existing CS-POGIL community of practice, and be invited to a one-day mid-year meeting. Finally, we will provide coaching and resources for instructors who wish to develop their own POGIL activities.

Objectives III and IV: Assess factors that affect faculty adoption and persistence with POGIL; assess the impact of using POGIL on student outcomes. These research objectives will involve a variety of data sources. *Direct video observation* of POGIL classrooms will enable us to observe and discover things that faculty and students might not notice or want to discuss in an interview. We hope to use these videos to develop a library of exemplary video cases. *Qualitative, semi-structured interviews* at the end of a POGIL course will enable us to explore faculty and student experiences, and gauge their opinions on POGIL and barriers to its implementation and the impact on student learning. *Surveys* provide quantitative evidence to complement qualitative measures. Faculty surveys will focus on their use of POGIL and their perceptions of its effects. Student surveys will focus on perceptions of learning, engagement, and critical thinking. Faculty and student interviews may identify emergent themes to inform additional survey items. *Student learning measures* will enable us to assess POGIL's impact on student learning.

IntroCS-POGIL Outcomes

The IntroCS-POGIL project is just starting. We are working to revise classroom materials, develop professional development sessions, pilot assessment instruments, and recruit faculty.

One promising new direction is a catalog of POGIL practices presented as pedagogical patterns [18, 19]. A pattern description typically contains several elements. The pattern's *name* should be concise and evocative. The *context* describes situations in which a pattern may be relevant. The *problem* statement is supported by a description of *forces* (positive and negative) that influence the problem. The *solution* statement is supported by a description of *consequences*, and potential responses. Pattern descriptions often include further *discussion* and *examples*, and refer to other related patterns. A pattern language for POGIL should help the POGIL community to more easily create, review, revise, and implement effective POGIL activities. A first paper described POGIL and some advantages of using patterns with POGIL, and described patterns for some of the structure and elements of POGIL activities, including models for learning cycles [20]. A second paper described patterns to help a teacher actively facilitate classroom learning [21].

Conclusions

This executive summary paper has described the motivation, objectives, activities, and outcomes for two NSF projects focused on POGIL in computer science. The CS-POGIL project developed sample POGIL activities for topics in intermediate level CS courses, and helped to develop a POGIL community. The IntroCS-POGIL project is a larger-scale study of how faculty implement POGIL in introductory CS courses and the factors that affect faculty implementation and student outcomes. Together, these projects are working to expand the set of POGIL activities for CS, the community of CS teachers who use POGIL, and the evidence of effectiveness.

These projects also explore some promising future directions:

- Tools for activity authors, including a *Design Canvas* to help sketch key activity elements, and an add-on for Google Docs to generate student and teacher versions of an activity from a single master copy.
- Integrating technology into activities, using a learning management system or tools like Jupyter Notebooks, which can combine text, live code, and other content.
- Several new professional development workshops for faculty, that focus on:
    - Writing and sequencing effective questions in an activity.
    - Identifying cultural bias, and making activities more culturally relevant.
    - Using patterns for activities and classroom facilitation.

Acknowledgements

This material is based upon work supported by US National Science Foundation grants DUE-1044679 (CS-POGIL) and DUE-1626765 (IntroCS-POGIL), a grant from the AAC&U TIDES Institute and the Leona M. and Harry B. Helmsley Charitable Trust, a Google CS4HS grant, and multiple Google CS Engagement grants.

References

1. A.W. Boykin, P. Noguera. 2011. *Creating the Opportunity to Learn: Moving from Research to Practice to Close the Achievement Gap*. Association for Supervision & Curriculum Development.
2. A.F. Chavez. 2011. Toward a multicultural ecology of teaching and learning: A critical review of theory and research. *Journal of Excellence in College Teaching*, 21, 4, 49-74.
3. G.D. Kuh, C.G. Schneider. 2008. *High-Impact Educational Practices: What They Are, Who Has Access to Them, and Why They Matter*. AAC&U.
4. E.P. Douglas, C. Chiu. 2013. Implementation of Process Oriented Guided Inquiry Learning (POGIL) in engineering. *Advances in Engineering Education*, 3, 3.
5. L. Lenz. 2015. Active Learning in a Math for Liberal Arts Classroom. *PRIMUS* 25, 279–296.
6. E.A. Mulligan. 2014. Use of a modified POGIL exercise to teach bacterial transformation in a microbiology course. *Journal of Microbiology & Biology Education*, 15, 1.
7. T. Vanags, K. Pammer, J. Brinker. 2013. Process-oriented guided-inquiry learning improves long-term retention of information. *Advances in Physiological Education*, 37, 233–241.
8. G. Jin, T.J. Bierma. 2011. Guided inquiry learning in environmental science. *Journal of Environmental Health*, 73, 6, 80-85.
9. T.W. Maurer. 2014. Teaching financial literacy with Process-Oriented Guided Inquiry Learning (POGIL). *Journal of Financial Education*, 40, 3/4, 140-163.
10. E. Mitchell, D. Hiatt. 2010. Using POGIL techniques in an information literacy curriculum. *The Journal of Academic Librarianship*, 36, 6, 539–542.
11. J.J. Farrell, R.S. Moog, J.N. Spencer. 1999. A guided-inquiry general chemistry course. *Journal of Chemical Education*, 76, 4, 570.
12. S.E. Lewis, J.E. Lewis. 2005. Departing from lectures: An evaluation of a peer-led guided inquiry alternative. *Journal of Chemical Education*, 82, 1, 135.
13. H.H. Hu, T.D. Shepherd. 2013. Using POGIL to help students learn to program. *ACM Transactions on Computing Education*, 13, 3, 13:1-13:23.
14. H.H. Hu, B. Avery. 2015. CS Principles with POGIL activities as a learning community. *Journal of Computing Sciences in Colleges*, 31, 2, 79-86.
15. S. Kumar, C. Wallace. 2014. Instruction in software project communication through guided inquiry and reflection. In *Proceedings of the IEEE Frontiers in Education Conference (FIE)*. (Madrid, Spain, Oct 22-25, 2014).
16. H.H. Hu, C. Kussmaul, B. Knaeble, C. Mayfield, A. Yadav. 2016. Results from a survey on faculty adoption of Process Oriented Guided Inquiry Learning (POGIL) in Computer Science. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (Arequipa, Peru, July 11-13, 2016).
17. J. Century, A. Cassata. 2014. Conceptual foundations for measuring the implementation of educational innovations. In L.M. Hagermoser, T.R. Kratochwill, eds, *Treatment integrity: A foundation for evidence-based practice in applied psychology* (pp. 81–108). APA.
18. J. Bergin. 2000. Fourteen Pedagogical Patterns. In *Proceedings of the European Conference on Pattern Languages of Programs (EuroPLoP)* (Irsee, Germany, July 5-9, 2000).
19. S. Fincher, I. Utting. 2002. Pedagogical patterns: Their place in the genre. *ACM SIGCSE Bulletin*, 34, 199–202.
20. C. Kussmaul. 2016. Patterns in classroom activities for Process Oriented Guided Inquiry Learning. *Proceedings of the Conference on Pattern Languages of Programs (PLoP)*, (Allerton Park, IL, Oct 24-26, 2016).

21. C. Kussmaul. 2017. Patterns in classroom facilitation for Process Oriented Guided Inquiry Learning. *Proceedings of the Nordic Conference on Pattern Languages of Programs (VikingPLoP)*, (Schleswig-Holstein, Germany, Mar 30 – Apr 2, 2017).