AC 2012-4593: PROCESS ORIENTED GUIDED INQUIRY LEARNING (POGIL) IN COMPUTER SCIENCE AND SOFTWARE ENGINEERING

Mr. Clifton L. Kussmaul, Muhlenberg College

Clifton 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 C.S. at Moravian College. He has a Ph.D. in computer science from the University of California, Davis, M.S. and M.A. from Dartmouth College, and B.S. and B.A. from Swarthmore College. His professional interests and activities include education, entrepreneurship, software engineering, digital signal processing, cognitive neuroscience, and music.

©American Society for Engineering Education, 2012
Process Oriented Guided Inquiry Learning (POGIL) for Computer Science & Software Engineering

Abstract

Educators have developed a wide variety of approaches to engage students, enhance learning, and emphasize attitudes and skills in addition to knowledge. One such approach is Process Oriented Guided Inquiry Learning (POGIL), in which teams of learners (typically 3-5) work on scripted inquiry activities and investigations designed to help them construct their own knowledge. The teams follow processes with specific roles, steps, and reports that help develop process skills and encourage individual responsibility and meta-cognition. POGIL has been developed and validated over the last 15 years, primarily in chemistry, and studies have found that POGIL significantly improves student performance.

POGIL has particular potential for education in computer science (CS) and software engineering (SE). Software development is largely a team-based, problem-solving activity, and POGIL helps students to develop team process skills and problem-solving abilities. POGIL also encourages students to collaborate and learn from each other rather than focusing on an instructor. POGIL in CS and SE also presents some challenges. There are not many POGIL activities for CS and SE, and developing them requires significant time and effort. CS and SE courses and curricula vary widely, and portions of the content change rapidly, making it more difficult to adapt or adopt materials developed elsewhere.

This paper describes an ongoing NSF funded project to develop POGIL activities for CS and SE. First, it reviews relevant background on effective learning and POGIL, compares POGIL to other forms of active learning, and describes the potential of POGIL for CS and SE. Second, it describes a sample POGIL activity for SE, including the structure and contents, student and facilitator actions during the activity, and how activities are designed. Third, it summarizes current progress and plans for the NSF project. Finally, it discusses student reactions, lessons learned, and future directions.

Introduction

To improve student learning, enthusiasm, and retention, especially in science, technology, engineering, and mathematics (STEM) areas, educators have developed a wide variety of active learning approaches to engage students, enhance learning, and emphasize attitudes and skills in addition to knowledge; a few reports are summarized below. Baldwin2 described experiences, benefits, and pitfalls with discovery learning, which broadly refers to learning through self-teaching. McConnell17 discussed active and collaborative learning (ACL), a set of ACL activities, associated risks and ways of addressing them, characteristics of and techniques to promote effective groups, and activity design tips. Gonzalez9 reported on CS1 sections where each session was roughly 1/3 discussion, 1/3 lecture, and 1/3 ACL, and students did significantly better in CS2 than peers from traditional sections. Beck and Chizhik3 reported a CS1 course where students spent roughly half of class on ACL exercises, and did significantly better than peers in a traditional section; that effect was found for a variety of majors and both genders. Sowell and colleagues20,21 described experiences with active learning in three courses, including
sample exercises, lessons learned, and qualitative and quantitative results; in general, student grades and course evaluations improved.

Process Oriented Guided Inquiry Learning (POGIL)

POGIL is based on learning science and shares characteristics with other forms of active, discovery, and inquiry-based learning. As described below, POGIL is a synergistic combination of effective learning practices. In POGIL, teams of learners (typically 3-5) work on scripted inquiry activities and investigations designed to help them construct their own knowledge, often by modeling the original processes of discovery and research. The teams follow processes with specific roles, steps, and reports that help students develop process skills and encourage individual responsibility and meta-cognition. The instructor serves as an active facilitator, not a lecturer or passive observer; effective facilitation is a key aspect of POGIL. POGIL activities and processes are designed to achieve specific learning objectives; typically an activity is designed to focus on 1-2 (disciplinary) concepts and 1-2 process skills. The POGIL Project (www.pogil.org) offers workshops to help faculty learn about POGIL theory and practice, including how to facilitate, evaluate, and develop POGIL activities.

POGIL activities generally follow a learning cycle with 3 phases. First, students explore a model or data that is provided or that they have collected, to find trends or patterns and generate and test hypotheses to help understand or explain the data. Second, students define or invent a new concept using the trends, patterns, or hypotheses; importantly, they have first constructed understanding. Finally, students apply the new concept in other situations or contexts to help generalize its meaning and applicability. Thus, the scripted activity provides information and asks questions to guide students through the learning cycle and help them develop process and learning skills.

Developing effective POGIL activities is time-consuming, but supporting resources are available. Generally, the author first identifies learning objectives and the focus of the activity. For examples, teams could analyze data, derive equations, or explore the behavior of a physical system. Next, the author creates a sequence of key questions to guide teams through the inquiry process. Finally, the author identifies and develops supporting information, such as prerequisites, glossary of terms, references, handouts, and subsequent assignments or projects. POGIL activities involve three types of key questions. Directed questions have definite answers, are based on material available to students, and provide a foundation for later parts of the activity. Convergent questions may have more than one answer, and require teams to analyze and synthesize information to reach non-obvious conclusions. Divergent questions are open-ended, do not have right or wrong answers, and may lead teams and individual students in different directions.

POGIL has been developed and validated over the last 15 years. POGIL has been used extensively in chemistry, and also in materials science and engineering, although POGIL is not yet well known in CS and SE. Multiple studies have found that POGIL significantly improves student performance. Table 1 below summarizes data from three experiments in different settings (but all in chemistry courses). In each case it compares grade distributions between cohorts of students in lecture-based and POGIL-based versions of a course.
Table 1: Summary of Experimental Data

<table>
<thead>
<tr>
<th># format</th>
<th>%A</th>
<th>%B</th>
<th>%C</th>
<th>%D,W,F</th>
<th>statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecture</td>
<td>19</td>
<td>33</td>
<td>26</td>
<td>22</td>
<td>$\chi^2=40.9$</td>
</tr>
<tr>
<td>POGIL</td>
<td>24</td>
<td>40</td>
<td>26</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>change</td>
<td>+5</td>
<td>+7</td>
<td>0</td>
<td>-12</td>
<td></td>
</tr>
<tr>
<td>lecture</td>
<td>20</td>
<td>20</td>
<td>27</td>
<td>33</td>
<td>$\chi^2=70.1$</td>
</tr>
<tr>
<td>POGIL</td>
<td>29</td>
<td>35</td>
<td>24</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>change</td>
<td>+9</td>
<td>+15</td>
<td>-3</td>
<td>-21</td>
<td></td>
</tr>
<tr>
<td>lecture</td>
<td>12</td>
<td>19</td>
<td>16</td>
<td>53</td>
<td>$\chi^2=19.1$</td>
</tr>
<tr>
<td>POGIL</td>
<td>9</td>
<td>32</td>
<td>31</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>change</td>
<td>-3</td>
<td>+13</td>
<td>+15</td>
<td>-25</td>
<td></td>
</tr>
</tbody>
</table>

POGIL for CS & SE

POGIL has particular potential for CS and SE education. Software development is largely a problem-solving activity since background information is usually available, and POGIL helps students to develop their problem-solving abilities. Teams are important to most IT organizations, and POGIL helps students to develop important team process skills. POGIL also encourages students to collaborate and learn from each other rather than focusing on an instructor. In chemistry and other disciplines, effective POGIL activities seem to be adopted and adapted by other instructors and at other institutions. Thus, POGIL may provide useful models for reusable materials in CS and SE education.

POGIL in CS and SE also presents some distinctive challenges. Currently, not many effective POGIL activities exist for these subjects; thus, faculty need to invest significant time and effort developing them. Courses and curricula are more varied than in chemistry, and portions of the content change more rapidly, making it more difficult to adopt or adapt materials at other institutions.

Sample Activity: Project Scheduling

This section describes activity Project Scheduling (PS) to show what a POGIL activity looks like and what the student teams and the facilitator do during the activity. It also describes common variations and some of the underlying motivations. The full activity is available online.

At the start of PS, the facilitator spends a minute or two providing motivation and background, and then distributes the activity to the student teams. Teams can be formed in many ways, e.g. at random, by ability (stronger students together, weaker students together), or by student choice. The following sections contain text from the activity (in italics), followed by elaboration and discussion. For brevity, some text has been simplified, and some sections have been omitted.

**Introduction**

*This activity will help you learn about project scheduling, including challenges & useful techniques. To do this, you will explore ways to estimate how long it will take to make a batch of 40-50 cookies.*
Before you start, assign a role to each member. If you have 4 people, combine Manager & Reflector. If you have 3, combine Speaker & Technician.

POGIL activities can have a variety of roles, but PS uses the following, which are common:

- **Manager**: keeps track of time and makes sure everyone contributes appropriately.
- **Recorder**: records all answers and questions, so team members have accurate notes.
- **Speaker**: speaks on behalf of the team to the facilitator, other teams, or the entire class.
- **Technician**: handles cards and worksheets (as well as computer or other lab equipment).
- **Reflector**: considers how the team could work and learn more effectively.

Roles provide multiple perspectives and motivate all team members to contribute. Students rotate roles for each activity, but teams usually stay together for a set of activities.

**Time Estimates for Recipe**

1. (1 min) Estimate **individually** (no discussion) how long it would take for **1 cook alone** to make a batch of cookies. Write each estimate below.
2. (1 min) Estimate **individually** (no discussion) how long it would take for **5 cooks together** to make a batch of cookies. Write each estimate below.
3. (2 min) Estimate **as a team** how long it would take for (a) **1 cook** (b) **5 cooks** to make a batch of cookies. Write your team estimates below.
4. (1 min) Summarize any insights or concerns raised in the discussion. Review progress with the facilitator before continuing.

Each team begins to work through the questions, which focus on the **exploration** phase of the learning cycle, and help direct student attention to the activity. Each question has an estimated duration to help teams manage time. The facilitator circulates around the room to monitor progress, clarify questions, and help address problems; teams are not expected to complete all questions in an activity unassisted. When most teams have reached question 4, the facilitator might ask teams to report their answers to question 3, and give each team a handout with the cookie recipe, and a set of cards containing steps in the recipe.

**Steps in Recipe**

5. (1 min) Sort the **16 step cards** into **5 groups**: Dry, Flavor, Wet, Bake, Cleanup
6. (3 min) Read through the cookie recipe provided in a handout. Note which steps are **explicitly** listed in the recipe and which are **hidden**. List 3-5 other steps that are in the recipe but not on cards.
7. (2 min) In the “**Group / Step**” column of **Scheduling Worksheet 1**, write an outline with all groups & steps. Put each group label in its own row, and put each step in that group into a row below the group row. Do not worry about the order of groups or steps - alphabetical order is fine.

This outline is called a **Work Breakdown Structure (WBS)**. A WBS can be useful to organize a project and identify missing steps, particularly if you have templates with common headings or WBS examples from similar projects. For complex projects, the WBS may have 3 or more levels and require multiple pages.

8. (2 min) Consider other grouping for the steps, and list their pros & cons. Review progress with the facilitator before continuing.
Question 5 is directed, since each card has a clearly labeled group; it directs teams to look at each card and become familiar with the steps in the recipe. Question 6 is convergent (most teams will have similar answers), but if there are notable differences the facilitator could initiate a brief discussion. Question 7 might be confusing; in some cases, teams will figure it out on their own; in other cases, the facilitator may provide assistance. Note that teams create a WBS before the terminology is introduced; this is an important aspect of the concept invention phase of the learning cycle. Question 8 is divergent (answers may vary), and provides an opportunity for faster teams to consider new ideas while slower teams catch up.

**Time Estimates for Steps**

9. (3 min) Divide the cards among team members. Individually estimate the time to complete each step and write the estimated time on each card.

10. (2 min) Add all of the step times: _____. Is this how long the recipe would take for 1 cook? Yes / No

Describe circumstances under which this would and would not be true.

11. (2 min) Divide the sum of step times by 5: _____. Is this how long the recipe would take for 5 cooks? Yes / No

Describe circumstances under which this would and would not be true.

12. (3 min) Which steps are most difficult to estimate? Why?

How could these estimates be improved? Summarize the discussion.

Review progress with the facilitator before continuing.

Questions 9, 10, and 11 are exploratory and convergent; again, the facilitator might ask teams to report and compare their answers. If teams produce significantly different schedules, the facilitator might encourage them to consult with each other to understand or resolve the differences. Question 12 is divergent, and could be skipped if time is short, or drawn out if some teams need more help with earlier questions.

**Dependencies between Steps**

13. (4 min) Assume there is just 1 cook, and organize the steps by placing the cards in order from left to right, minimizing the total time the recipe will take.

Review progress with the facilitator before continuing.

14. (3 min) Copy the schedule onto Scheduling Worksheet 1, with a bar showing the start & stop time for each step. Hint: Check the total time and adjust the horizontal axis or get more sheets if necessary.

15. (2 min) When there are many steps, it can be useful to look at groups. What is a logical start and stop time for a group of steps? Add these to the worksheet as bars. This is a Gantt chart; it was developed in the early 1900s by Henry Gantt.

16. (3 min) How could the Gantt chart show each step’s status (e.g. not started, in progress, finished)? How useful would this be to review the overall project status? What other data might be useful? Review progress with the facilitator before continuing.

Questions 13, 14, and 15 focus on concept invention, and precede introduction of new terminology. In question 13, teams organize the schedule graphically, which is easier for them to adjust and for the facilitator to review, before transcribing the schedule onto a worksheet in question 14. In one sense, questions 15 and 16 prompt teams to apply what they have learned.
to interpret the Gantt chart, but in another sense they prompt teams to *invent* new concepts that could be applied later. (Such overlap is common.)

This activity could be adapted to use a spreadsheet instead of paper worksheets. Students unfamiliar with spreadsheets could learn from their teammates; if an entire team needs help, the facilitator could work with them, or ask a student from another team to assist them.

The next sections of the activity (not shown here) extends the previous concepts to multiple cooks and to shared resources (such as a food processor) that might also need to be scheduled, and then to consider which steps can and cannot be shifted time, leading to the concept of a critical path. Thus, teams must *apply* their earlier understanding to *invent* extensions or variations.

Next Steps

Teams usually finish the entire POGIL activity during class, where the facilitator is available to monitor progress and provide assistance if necessary. In some cases, teams may need to complete an activity outside of class.

A POGIL activity is usually followed by assignments, assessments, or other tasks to help solidify and/or assess student learning. These might include:

- Traditional homework assignments to design, develop, test, or enhance software.
- A short quiz during the next class meeting.
- A written report summarizing what the team did and learned in the activity; this might be drafted by the Recorder and reviewed by other team members.
- An individual peer assessment to help identify and correct problems within a team.

Project Plan

This section summarizes current progress and plans for a project to develop POGIL materials for CS and SE, supported by the National Science Foundation (NSF) Transforming Undergraduate Education in STEM (TUES) program. This project leverages the successful POGIL approach, extends it to CS and SE, and explores some new directions that could benefit POGIL in other fields. Specifically, this project develops, refines, validates, and disseminates two sets of 8-10 POGIL activities for CS & SE. Set A focuses on data structures & algorithms (DS&A), including queues & stacks, lists, hash tables, recursion, searching, and sorting. Set B focuses on software engineering including project scheduling (e.g. work breakdown structures, Gantt charts, critical path), estimation (e.g. story points), project budgeting and financial planning, and elements of the unified modeling language (UML). These activities are packaged as a set of handouts containing background information and questions, including space to write answers. Each activity packet contains appropriate supporting materials, such as worksheets, code samples, references, and rubrics. Each packet also contains overview information for facilitators, such as learning objectives, prerequisites, directions to prepare the activity, notes for facilitating, a brief history of the activity, and planned or suggested improvements.

As the activities are developed, they are reviewed by the Co-PIs, who are chemistry faculty with extensive POGIL experience, and by CS and SE faculty collaborators who have completed
introductory POGIL workshops and plan to use POGIL in their classrooms. An assessment 
expert guides and supports more detailed assessment plans and activities. As activities are 
refined and validated, they are disseminated to the broader CS and SE education community 
through papers, conference presentations, workshops, and a project website\textsuperscript{14}.

Thus, this project seeks to produce materials for adoption by other CS and SE faculty, and 
thereby improve the quality of CS education, and particularly for average and below average 
students. The project also seeks to identify and foster a POGIL community within CS. 
Table 2 summarizes the project timeline.

\begin{table}[h]
\centering
\caption{Project Timeline}
\label{tab:project_timeline}
\begin{tabular}{|c|c|c|}
\hline
\textbf{Timeframe} & \textbf{PI \& co-PIs} & \textbf{Faculty Collaborators} \\
\hline
2011 Summer & plan assessment & POGIL training \\
& create/refine PAs & review PAs \\
\hline
2011-12 AY & pilot/refine PAs & review/pilot PAs \\
& show: talks & evaluate (baseline) \\
\hline
2012 Summer & create/refine PAs & review/refine PAs \\
\hline
2012-13 AY & refine PAs & evaluate \\
& push: workshops & (baseline/treatment) \\
\hline
2013 Summer & create/refine PAs & create/refine PAs \\
\hline
2013-14 AY & refine PAs & evaluate (treatment) \\
& push: talks/workshops & pilot/refine PAs \\
\hline
\end{tabular}
\end{table}

**Project Status**

Currently, the following POGIL activities have been drafted, piloted, and revised\textsuperscript{14}:

A. **Data Structures \& Algorithms**
   1. Searching (linear \& binary)
   2. Queues \& Stacks
   3. Linked Lists
   4. Sorting (mergesort \& quicksort)
   5. Maps \& Hash Tables
   6. Sequence Analysis (bioinformatics)

B. **Software Engineering**
   1. Teams \& Roles (introductory)
   2. Project Scheduling (PERT, critical path)
   3. UML Use Case \& Activity Diagrams
   4. UML Data Diagrams \& CRC Cards

C. **Programming Constructs \& Techniques**
   1. Unit Testing (JUnit)
   2. Code Reading
   3. Error Handling \& Exceptions
   4. Reading \& Writing Files
   5. Inheritance
The activities in A and B would require only minor changes for other languages. The activities in group C could be translated for other programming languages.

Assessment Plan

Table 3 summarizes the project assessment plan, which will be revised and expanded during 2011-2012, with assistance from an assessment expert. SIR-II is the Student Instructional Report II™, and TDS is the Team Diagnostic Survey.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Techniques(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Students</strong></td>
<td></td>
</tr>
<tr>
<td>Improve learning outcomes.</td>
<td>Average grades.</td>
</tr>
<tr>
<td></td>
<td>Qualitative assessment of selected assignments.</td>
</tr>
<tr>
<td>Improve affective outcomes.</td>
<td>Current (e.g. SIR-II™), existing (e.g. TDS™),</td>
</tr>
<tr>
<td></td>
<td>&amp; custom instruments.</td>
</tr>
<tr>
<td>Improve recruiting &amp; retention.</td>
<td>Course enrollments &amp; major/minor counts.</td>
</tr>
<tr>
<td><strong>PIs &amp; Project Team</strong></td>
<td></td>
</tr>
<tr>
<td>Develop &amp; refine PAs.</td>
<td>Quarterly activity reports, peer review, interviews.</td>
</tr>
<tr>
<td>Improve faculty affective outcomes.</td>
<td>Reflection, interviews.</td>
</tr>
<tr>
<td>Enhance PAs (e.g. with Free &amp; Open</td>
<td>Quarterly activity reports, peer review, interviews.</td>
</tr>
<tr>
<td>Source Software).</td>
<td></td>
</tr>
<tr>
<td>Raise awareness (talks &amp; workshops)</td>
<td>Quarterly activity reports, interviews.</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
</tr>
<tr>
<td>Foster community for POGIL in CS.</td>
<td>Talk/workshop attendance &amp; evaluation forms.</td>
</tr>
</tbody>
</table>

Student Feedback

During 2009-2010 the author developed POGIL and other active learning activities for a graduate course (roughly 20 students) on genetic algorithms, neural networks, and fuzzy systems. The activities were presented as slides which were later posted for students to review. This reduced paper use and made it easier to tweak activities during class by adding steps or providing more information, and to manage the activity pace.

At the end of the course, students were asked to describe their experiences. The responses included recurring themes. Students were initially uncertain about an unfamiliar teaching and learning style, but then realized that they had mastered concepts and acquired useful process skills. Some students also had some difficulty understanding the instructor’s spoken language; however, individual teams could discuss the activity questions in any language(s) they preferred. For example:

*Group discussions & exchanging ideas with other groups is a better one. This makes us think about it in a better way. As you come to each group to talk, we can clarify our doubts by discussing it with you. And you encourage us to think more about the topic.*
Conclusions

POGIL is based on learning science and has a proven track record in other disciplines. There are a variety of materials to help faculty develop and improve POGIL materials. POGIL classrooms are very different from lecture-based classrooms, but POGIL shares characteristics with other forms of active, discovery, and inquiry-based learning, so that faculty familiar with such approaches should not have difficulty adapting to POGIL.

Future Directions

Future directions for this work include developing and revising more POGIL activities, extending and adapting them for courses and faculty at other institutions (including high schools and community colleges), developing POGIL activities in other areas related to CS and SE, and working to develop a community of CS and SE faculty using POGIL. We also hope to explore the potential of tools, such as wikis (e.g. TikiWiki) and learning management systems (e.g. Moodle), to help scale activities to larger courses and a broader community. Effective use of such tools could help the facilitator to monitor team progress, and enable student teams to more readily summarize and compare their results and conclusions.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant DUE-1044679. Any opinions, findings & conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. This material was also supported by a Fulbright-Nehru teaching award from the US-India Educational Foundation (USIEF).

Bibliography

   *Journal of Chemical Education*. 76(4):570.
   In *Proceedings of the SIGCSE Technical Symposium on Computer Science Education*. ACM, 133–137. 
   DOI= http://doi.acm.org/10.1145/1121341.1121386.
    eds. *Faculty Guidebook - A Comprehensive Tool for Improving Faculty Performance*. 2nd ed. Pacific Crest.
    *Journal of Chemical Education*. 77(1):120.
    Oxford University Press.
    The active-learning transformation: a case study in software development and systems software courses. 