Process Oriented Guided Inquiry Learning in Computer Science

Clif Kussmaul, PhD
Associate Professor of CS, Muhlenberg College
clif@kussmaul.org kussmaul@muhlenberg.edu

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.

POGIL = Process Oriented Guided Inquiry Learning

- Learners work in **teams** (of 3-5) on **scripted activities** that guide them through **inquiries** (investigations) to help **construct knowledge**.
- Teams follow **processes** with specific roles, reports, etc.
- Activities & processes are designed to achieve **learning objectives**.

POGIL evolved over many years.

- David Hanson, Stony Brook University, 1994
- Rick Moog, Franklin & Marshall
- series of 20+ NSF grants
- originally in chemistry, spreading elsewhere
- regular training workshops
- useful resources & active community:
  http://www.pogil.org

There is evidence that POGIL works.

#	format	%A	%B	%C	%D,F,W
1	lecture	19	33	26	22
x2=41 α< 0.005	POGIL	24	40	26	10
	change	**+5**	**+7**	**0**	**-12**
2	lecture	20	20	27	33
x2=70 α< 0.01	POGIL	29	35	24	12
	change	**+9**	**+15**	**-3**	**-21**
3	lecture	12	19	16	53
x2=19 α< 0.005	POGIL	9	32	31	28
	change	**-3**	**+13**	**+15**	**-25**

3 sample studies

- lower student attrition
- improved content mastery
- improved learning skills
- better attitude & motivation

Typical Activity Startup

1. Form teams of 3-4 people with roles:
   - manager, recorder, speaker, reflector
2. Fill out header of meeting minutes.
3. Start to work through activity.
   1. Raise your hand if you have questions.
   2. If I raise my hand, raise yours & be quiet.

Genetic Algorithms: Solutions

- Genetic algorithms (GAs) encode **solutions** as sequences of symbols from an **alphabet**.
  01101 ABDBABCA ZYBA

For each **solution** above:

1. What is L? (length of solution)
2. What is s? (size of alphabet)

Genetic Algorithms: Schemata

- When we analyze & discuss GAs, we often describe patterns called **schemata**.
- In each **schema**, some values are required, while others are **wildcards**, shown by *.

  1*0*   1**0**0*   AA**DC

For each **schema** above:

1. What is L? (length of schema)
2. What is s? (alphabet size)

GA Schemata: Counting

1*0*   1**0**0*   AA**DC

1. For each schema above, list all matching solutions.
2. Using L & s, how many solutions are there for a schema with w wildcard values?
3. How many schemata match a given solution?
4. How many schemata are possible, total?

GA Selection: Questions

- **A(t)** is the **population** (solution set) at time **t**
- **f(a)** is the **fitness** of solution a ∊ A(t)
- **m(a,t)** is the **count** of **a** at time **t**

1. What is the **total fitness** of A(t)?
2. What is the **average fitness** of A(t)?
3. What is **m(a,t+1)**, the count at time **t+1**?
4. Explain #3 and how it could be used (use complete sentences).

GA Selection: Analysis

- A(t) is the **population** at time t
  - population size |A(t)| is often constant
- f(a) is the **fitness** of solution a ∊A(t)
- m(a,t) is the **count** of solution a at time t

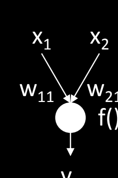
$$m(a,t+1) = (\text{population size})(\text{probability of a})$$
$$= |A(t)| \frac{f(a)}{\sum_{A(t)} f(a_j)} = \frac{f(a)}{\frac{1}{|A(t)|}\sum_{A(t)} f(a_j)} = \frac{f(a)}{f(A(t))}$$

Artificial Neural Networks (ANNs)

- ANNs are a **classification technique** inspired by natural neural systems.
- ANNs contain **neurons** (**input, output,** & **hidden**)
- Each neuron has **inputs** (with **weights**), which combine to produce an **output** via an **activation (or transfer) function**.
- Reacting (forward) is easy, learning (backward) is harder.

ANN: Perceptrons

- Simple model for ANNs
- Inputs $x_1, x_2, \ldots, x_n$
  e.g. 0..1, -1..1, -½..+½
- Weights $w_{11}, w_{12}, \ldots, w_{nm}$
- Outputs $y_1, y_2, \ldots, y_m$
  e.g. 0..1, -1..1, -½..+½
- Activation function f()



$$y_j = f\left(\sum_{i=1..n} x_i w_{ij}\right)$$

ANN Design: Questions (1)

Consider a neuron with **step activation function**, and where **-1=false** and **+1=true**.

1. Choose weights to compute:
   a. logical NOT (1 input)
   b. logical OR (2+ inputs)
   c. logical AND (2+ inputs)
2. Can one activation function serve a, b, & c? Explain why or why not.

ANN Design: Questions (2)

1. On an (x_1,x_2) graph, draw lines for $y = x_1w_1 + x_2w_2$ for w_1 & w_2, y={-1, 0, +1}
2. How would activation function **f** affect the lines drawn in #1?
3. Explain what happens with 3 or more inputs.

ANN Design: Questions (3)

1. Explain the issues that arise when you try to choose weights to compute XOR.
2. In complete sentences, explain the pros & cons of **one-layer networks**.
3. Choose weights to compute XOR with a two-layer network.
4. In complete sentences, explain the pros & cons of **multi-layer networks**.

ANN Design: Questions (4)

Consider a 3x3 binary image (see above).

1. Design a network & weights where each pattern above has a **unique output**.
2. How could you use **fewer neurons**?
3. How could you allow **noisy data**? (e.g. 1-2 incorrect pixels)

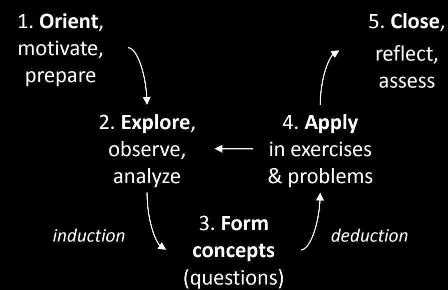
People learn better when they:

- work in teams
- combine content & process
- construct knowledge
- follow learning cycles
- connect concepts & representations
- receive prompt, regular feedback
- reflect on process & progress

Teams & processes enable students to learn from each other.

- Students teams learn, understand, & remember more.
- Processes provide helpful scaffolding.
- Students also learn process skills, such as communication & teamwork.
- Students can answer each others' questions; teacher answers more difficult questions.
- The best way to learn is to teach.

Activities are designed with stages to form a **learning cycle** (Karplus, Piaget).



Questions & problem solving move from simple to complex issues.

1. **Directed** - Develop context & confidence.
    1. Count, summarize data.
2. **Convergent** - Lead toward concept formation.
    1. Derive expression, synthesize concept.
3. **Divergent** - Encourage future exploration.
    1. Extrapolate, consider applications, broader impact.

Reporting & meta-cognition help students learn how to learn better.

- Reports help learners & teachers see & think about what happened in team.
- Getting learners to think about their learning is key to becoming better learners.

Responsibility & evaluation provide incentives & avoid problems.

- Each student has a role, and roles rotate.
- Students know they depend on each other.
- Free-loading is discouraged.

- Depending on the academic environment, reports may or may not be graded.

POGIL has 7(+1) key components.

1. Learning teams
2. Guided-inquiry activities
3. Questions to promote critical thinking
4. Problem solving
5. Reporting
6. Meta-cognition
7. Individual responsibility
8. Grades (when necessary)

POGIL has variations, pros & cons.

- Small & large courses (scales reasonably well).
- Daily, weekly, or occasionally.
- Paper and/or technology for activities & reporting.
- Adapts to varied languages & cultures.
- Harder to predict & control class.
  "guide on the side" vs. "sage on the stage"
- Requires time & effort to design activities.

My experiences with POGIL are recent & exciting.

- 2009: POGIL workshop at ASEE
- 2009-10: POGIL for soft computing, India
- 2010-11: regional & national POGIL meetings
- 2010-: POGIL for other CS topics
  - software engineering, DS&A
- 2011-: NSF TUES grant for POGIL in CS
  - looking for reviewers & collaborators…

Clif Kussmaul

In India, POGIL for soft computing went (surprisingly) well.

- 2009-2010 Fulbright-Nehru Scholar, India
- ~18 master's students
- POGIL was a big change from lectures
  - skeptical, then enthusiastic
- activities presented using PowerPoint
  - less paper, more flexible, manage pace & "reveal"
- fluent in English, prefer local language(s)
  - reduced potential language barriers

Clif Kussmaul

CS-POGIL: NSF TUES (Type I)

- Goal 1: POGIL activities for CS
  - Data Structures & Algorithms
    - stacks, queues, lists, hashing
    - searching, sorting
  - Software Engineering & Project Mgmt
    - project scheduling, estimation
    - Unified Modeling Language
- Goal 2: Foster CS-POGIL community.

Clif Kussmaul

CS-POGIL: Current Status

- Content
  - Project Scheduling
  - Linked Lists
  - Queues & Stacks
  - Sorting
  - Searching
  - JUnit Testing
  - Java Exceptions
  - Java Inheritance
- Talks & Workshops
  - US & India
- Community
  - CS faculty
  - chemistry faculty
  - high school teachers
  - student researchers

Clif Kussmaul

CS-POGIL: Future Directions

- web-based activities
- Moodle enhancements
- bridge to other disciplines
  - e.g. engineering, business
- POGIL patterns

- most involve student researchers

Clif Kussmaul

POGIL Resources

- Fri: poster — Helen Hu
- Sat: special session — Helen Hu & Clif Kussmaul

- Clif Kussmaul — clif@kussmaul.org, http://cspogil.org
- The POGIL Project — http://pogil.org
  - Hanson. Instructor's Guide to POGIL.
  - Hanson. Designing POGIL Activities.
- July 2012: Regional POGIL Workshops
  - invaluable if you want to get started

- Eberlein, Kampmeier, Minderhout, et al. (2008) Pedagogies of engagement in science: A comparison of PBL, POGIL, and PLTL. *Biochemistry & Mol. Bio. Education* 36(4):262–273