Research, Teaching & Service:
The Miniconference as a Model for CS Graduate Seminar Courses

Paul Sivilotti
Bruce Weide
Computer Science & Engineering
The Ohio State University
(paolo,weide)@cis.ohio-state.edu

What keeps me up at night
- 3 graduating PhD students on job market:
  Scott Pike - Fault Containment
  Nigamanth Sridhar - Design Patterns
  Chris Bohn - Model Checking
- 2 other affiliated students on job market:
  Murat Demirbas - Sensor Networks
  Jason Hallstrom - Software Product Lines

Do they know what they are getting into?

Inadequacy in Preparation
Each level requires different skills:
- High school
  - take tests
- Undergraduate student
  - answer questions
- Graduate student
  - ask questions
- Faculty member
  - run a business

Our Idea
- Distill the "academic experience"
  - Create a course structure that captures all the essential elements
- Constraints
  - 1 course (10 weeks)
  - No sacrifice of technical content
  - Heterogeneity of audience
- We're bold, but also realistic
  - Preparing future faculty is a daunting task
  - Programs, workshops, panels, seminars, books, ...
  - Our course structure is just one small step

The Miniconference Model
- During the term
  - Professor covers normal technical material
  - Students make some seminar presentations
  - Students carry out original research projects
- Seminar culminates in a "miniconference"
  - Call for papers is circulated
  - Papers are written and submitted
  - The class acts as the program committee!
    - Review & critique papers
    - Make accept/reject decisions
    - Accepted papers are presented

Past Miniconferences
- Sample projects:
  - Dynamic interceptor composition framework
  - Temporal component-based specifications
  - Dynamic software module replacement
  - Heuristics for distributed scheduling
  - Distributed discrete-event simulation app
  - Structured parallel programming techniques
  - Distributed recording service for debugging
  - Encapsulating concurrency for sequential reasoning
- Many accepted papers led to "real" publications
The Three Pillars

This model touches on all three aspects of academia:
- **Research**
  - Focused, graduate-level research project
  - Miniconference paper preparation & submission
  - Miniconference paper presentation
- **Teaching**
  - Seminar presentation to class
  - Topic chosen to relate to student’s project
- **Service**
  - Professional service: paper reviewing
  - Professional service: program committee

Mechanics: Grading Scheme

- **Research**
  - 55%
- **Teaching**
  - 25%
- **Service**
  - 10%
- **Class Participation**
  - 10%

Mechanics: Paper Review

- **Review metrics**: specific and quantitative
  - **A. Relevance**
    - Category, audience, appropriateness
  - **B. Presentation**
    - 14 different metrics, scale of 1-10
  - **C. Contribution**
    - Importance, strengths, weaknesses, correctness
  - **D. Conclusions**
    - Overall recommendation, confidence level
  - **E. Open-ended (private and public) comments**
    - Everyone reviews several (~3) papers

Benefits of This Model

- Many similar approaches exist, with similar benefits
  - Writing skills
  - Oral presentation skills
  - Critiquing skills
- In addition, miniconference model reveals:
  - Insights into each of the “three pillars”
  - Inter-relationships among them

Learning Gains: Research

- Writing a good research paper
  - Targeting a specific audience
  - Packaging the paper’s contribution
- Giving a good research talk
  - Breadth and interests of audience
- Collaborative research and writing
  - Dynamics of writing a joint paper
- Mentoring junior students
  - In student-led projects, senior students can play the role of advisor
**Learning Gains: Teaching**
- **Potential synergies** between research and teaching
  - Experience with research project improves student lectures
  - Preparing for lectures helps with project
- **Contrast** research talk and lecture
  - One is meant to instruct, the other to inform (and persuade, and sell)
  - Different evaluation forms used

**Learning Gains: Service**
- **Reviewing and critiquing peers’ work**
  - Authors see (anonymous) reviews
  - PC committee sees all (other) reviews
  - Note: PC committee consists (only) of authors!
- **Behind-the-scenes look at PC decision process**
  - Each submission discussed to reach consensus
  - No one has seen every paper
  - Outcome can be influenced by the right “champion”
  - Seriousness: outcome affects people’s lives
  - Conflicts of interest: discussing competitors’ papers

**Robustness of the Model**
- We have tried many variations
  - Low vs high miniconference acceptance rates
  - Group vs individual projects
  - Professor- vs student-led projects
  - Long vs short revision windows
  - Relative weightings in grade distribution
- Other things we have not changed (yet)
  - Graduate-level seminar courses
  - Single-course scope for miniconference

In general, the model is surprisingly robust!

**Pointers for Success**
- **Specific, focused research projects**
- **Authentic CFP:**
  - Firm deadline, page limits
  - Include typesetting requirements (ACM)
- **Authentic conference:**
  - Session chair imposing time limits
  - “Published” proceedings
- **Focused quantified review templates**
- **End-of-term debriefing session**

**Summary**
The miniconference model for graduate seminars:
- Provides a microcosm of academia
- Introduces inter-relationships among research, teaching, and service
- Is surprisingly robust under variation
- Is worth trying!
What else might one add to this model?
- Proposal writing and funding (?)
- Promotion and tenure (???)
- Department/university citizenship
- The joy of advising students!

**What keeps me up at night**
- 3 graduating PhD students on job market:
  - Scott Pike - Fault Containment
  - Nigamandh Sridhar - Design Patterns
  - Chris Bohn - Model Checking
- 2 other affiliated students on job market:
  - Murat Demirbas - Sensor Networks
  - Jason Hallstrom - Software Product Lines

They may be better prepared than most
Research, Teaching & Service:
The Miniconference as a Model
for CS Graduate Seminar Courses

Paul Sivilotti
Bruce Weide

Computer Science & Engineering
The Ohio State University

(paolo.weide)@cis.ohio-state.edu