How to Write a CS Paper: What Your English Teacher Didn't Cover and Doesn't Know

Randall Bramley

Indiana University

13 September 2004

Work supported in part by NSF Grants CDA-9601632 and EIA-0202048 and DOE SciDAC Program

Overview

- Getting a topic to cover
- Mechanics of writing a CS research or project paper
- Politics of publication
- Presentation of the work

Finding the Project Topic

- Any unanswered question can turn into a research topic
  - More interesting ones are obviously better bets
  - Even a dull one can lead to surprises
    - Numerical algorithm for solving linear inequalities had the question "*what is computational order of your method*?" Answer: polynomial time … making it an extraordinary breakthrough
  - Never trust common sense – not yours and certainly not that of others
    - I/O directly to a file is faster than calling the same I/O functions indirectly through a library, of course.
  - Never give or take an answer that can be easily checked with a moment's coding
    - For any question there is an answer that is simple, direct, and wrong

Developing the Project Topic

- *Serialization* is process of turning an internal programming object into a linear stream of bytes suitable for writing to a file or network connection
- Java 1.0 claimed to have serialization that took linear time with respect to object size … but testing showed quadratic time.
- Java uses a *hash table* to track what's been serialized; what happens when a hash table gets full?

Developing the Project Topic

- Don’t confuse theory and reality - but The Gap is a great source for new research problems
  - Wonderful theorem may require new implementation
  - Interesting computational observation may require new theory
- Wilkinson (1946) solved linear system of 16 eqns by hand.
  - Common Sense: for problems of that size, rounding errors would lead to zero digits of accuracy in solution.
  - Observation: it was accurate to next to last digit
  - Led to an entire area of research (backward error analysis)
- Bramley (1989) realized we can solve large linear least squares problems accurately … giving a Ph.D. and ultimately tenure (row projection methods, Stokes problems in CFD, model reduction in constrained optimization)

Possible Systems Topics

- Time all of your computations
  - Where is the time being taken?
  - Why is that part the bottleneck?
  - Can the bottleneck be removed by a better algorithm or implementation?
- Apply a technique from one area to a another area.
  - Consider taking measurement of CPU utilization from a machine every second for x years. Can we apply signal processing statistics to analyze a “signal” that can then be used to predict future CPU utilization?
  - Parallel computing involves partitioning large graphs to get node clusters with few inter-node edges. Can we use those techniques to reorganize files on a hard drive to minimize jumping around the HD?

Paper Writing Mechanics

- Every paper/thesis/research project report has the same *structural* outline
  - A clear statement of the problem
  - A survey of what has been done by others in the field, with an identification of what is lacking
  - A statement of what you have done/contributed that is new and different, and how it solves the problem
- Always start project with an *intellectual* outline:
  - Three major points to make or questions to address or ideas to explore
  - Define tests to validate or explore the points/questions
  - Title, abstract, introduction, conclusions are written *last*
  - The outline and ideas will evolve as the work proceeds!

Writing Mechanics

- Issue not to be underestimated: *what text processing language*?
  - Latex is best in world for mathematical text, equations
  - Bibtex works with Latex, gives flexible, great-looking bibliography that can be reused for future papers.
  - Latex/bibtex can be used with CVS to track changes, notify others
  - MS Word is more familiar to most people
  - MS Word has change tracking, valuable for late-stage changes and visual identification of who made them. But format is binary, so CVS is useless
  - Both Latex and Word are cranky and difficult for large documents (thesis or book). Framemaker is better tool then.
  - Conferences/journals often supply a format, always give requirements for margins, etc. Use those from the start!

Writing Mechanics: General

- Use top-down, stepwise refinement method starting from the outline.
- Multiple authors: agree from first on person, tense, tone.
- Multiple authors: have one clearly identified lead author for final merge, formating, and submission
- Never put off implementations or testing until last: it *always* will lead to changes in the paper, is *never* predictable.
- Every claim in a paper must be backed up in one of two ways:
  - With a citation to previously published working
  - With a proof of the theorem, or adequate testing results, to substantiate the claim.
  - Scale back your claims if necessary to match what was proven

Writing Mechanics: Testing

- State testing conditions well enough to reproduce them
  - Hardware, memory configuration (cache and main), hard drive speed, cpu model, etc. (Look at *sysinfo* command on Solaris)
  - OS name and version number
  - Language(s) used, compiler name, version, compilation flags
  - Job launch mechanism if relevant
  - Load on machine; shared resource or dedicated machine?
  - File system, hard drive transfer rates, libraries used
- Give statistics: multiple runs with average and standard deviation. Show timings with error bars at +/- one std dev.
- Discard outliers only if you state you've done so in the paper

Writing Mechanics: Testing

- Use a standard benchmark if one exists: for compiler optimizations, CPU performance, I/O rates such exist
- If no standard exists, try to use tests that others have published and used – especially if you beat them. Don't cherry pick tests.
- If no standard exists, consider proposing your set of tests as a standard benchmark
- Only create new standards if a need exists – and be ready to give a lively defense for the new benchmarks
- Provide your code and tests as open source, freely available code and data.

Side Note: Bramley's Rules for Committees:

- Last slide almost violates one of the BRC
- Never go to a committee meeting unless an agenda posted in advance
- Never go to a committee meeting unless the end time is posted
- Never serve on a committee with more than four others
- Never serve on a committee with people who don't have a direct interest in the success of the committee's work
- Only serve on committees with a finite lifespan and an absolute deadline for getting the work done
- Never serve on a committee with the words "standards" or "policy" in the title

Bibliography

- Always have a bibliography, it's how you spread credit/blame
  - Citations are the coin of the realm in academia
- Make sure you have read every paper cited in the bibliography
  - Don't use a ref from another paper without checking it yourself
- URL's are convenient, but prefer long-lasting/more stable publications like journals, tech reports, hard copy.
  - Typical phrasing: "Also see URL http://..."
- When an idea is common in the area, try to cite the *first* person to have published it. Exception: a really good survey paper.
- Some ideas are folk knowledge: no citation needed, e.g. the halting problem.

Bibliography

- Journals and conferences will give their format guidelines
  - Citations numbered and in brackets [3,7]
  - Citations given as author-year parentheses (Springer1980)
  - Citations given as numbered trailing superscripts[3,7,8]
  - Journal name in italics, article in roman letters … or vice-versa?
- Some people are anal-retentive about bibliographies!
  - Lop off one page from their paper (pp. 36-67) and they'll have a cow

The Politics of the CS Paper

- Multiple authored papers lead to greatest unsolved problem in Computer Science today: order of author's names.
- As a student, always list your faculty advisor first
  - Especially if they pay your salary
  - A good advisor will change the order to put you first if major portion of ideas and work was your
  - There are bad advisors out there (but not at IU, *of course*.)
- If approx equal contributions made, list names alphabetically
- If someone insists on grabbing more credit than is their due, don't write future papers with them
  - Corollary: may have to get another advisor

The Politics of the Paper

- Venues for a paper are refereed or not refereed. Also called “peer reviewed” papers.
  - 3-5 reviewers who read paper, give anonymous report to editor
  - There’s bad reviewers out there (but not at IU, *of course*.)
  - Main unrefereed paper venue: departmental technical reports
- Where do editors get reviewers?
  - From their friends and enemies
  - From the authors listed in your bibliography, especially if you claim to have topped someone with a better method, implementation, etc.
  - Corollary: you can sometimes use bibliography entries to help steer the paper into the hands of friendly souls.
  - Good researchers *like* being beaten, it improves the area
  - Never get into a pissing match with reviewers; they have the power

The Politics of the Paper

- Most refereed systems and beginning student papers go to conferences.
  - Conferences have a tighter schedule than journals
  - With a journal, multiple iterations between referees and authors possible
  - With a conference, you get one shot, no chance to rebut a reviewer
  - Means you must *anticipate* possible complaints
- Conference or journal, pick up latest issues or proceedings of it and read for style, topics, emphasis.
- Three directions in general a researcher can go
  - With the herd, validating the existing paradigm
  - Iconoclastic breaking of sacred beliefs
  - Off in a new direction, looking at problems no one else has
  - Guess which one leads to easiest acceptance of a paper
  - Guess which one leads to greatest self-esteem and recognition

The Politics of the Paper

- You can never publish negative results, unless cast as a positive result.
- Papers often have a "future work" section, which consists of the author marking ideas as his own without actually having to pursue them or do any real work.
- LPU's: least publishable units to maximize number of publications
  - Encouraged by bean counters at companies and labs
  - Encouraged by draconian, inflexible page limit rules of publishers

Presenting Your Paper

- What is the time limit for your talk?
  - 20 minutes, have two main points you want to make
  - 30-50 minutes, have three main points you make
  - Finish five minutes late, everyone hates you
  - Finish five minutes early, everyone loves you
  - Insert extra transparencies or hidden PPT slides so you can fill out extra time or answer predictable questions
  - Have some slides identified as skippable, if running out of time
  - Put a timing note on bottom/top or hidden on slide so you'll know where you should be at that point ("5 min")
  - Always leave time for questions, it's rude otherwise

Presenting Your Paper

- Don’t just give the paper, with same graphs, etc.
- Talk is your chance to convey things not allowed in paper
  - Results that were suggestive but not proof of an idea
  - Negative results that referees would not allow
  - A narrative story is more natural for a talk
- Can also give more recent results since paper was written
  - Journals have 1-3 year lag time between submission and publication(!)
  - Conferences typically have 3-12 month lag time

Presenting Your Paper

- The dreaded question period of a talk
  - Genuinely puzzled speakers may find it hard to articulate the question
  - Give an answer, then ask them “does that answer your question?”
  - Often can get valuable references, citations, related ideas from questions
    - An irrelevant question suggests you need to clarify the talk
  - Questioners who really want to give their own lecture
    - Let them run down, smile, and ask “uh, what was your *question*?”
  - It’s ok to say “I don’t know”, never try to BS your way through
  - It’s ok to say “I’ll find out whether or not we used JIT for those numbers, and send you the answer”. Be sure you do so.
  - Handle even hostile questioners with a smile and good nature – you’ll win far more points than they will ultimately

Presentation of the Paper

- MS Power Point (or Open Office Impress) are now the standard means of presentation. Learn to use them effectively!
- Be prepared for failures! Can you give a decent talk if reduced to a chalkboard? This is a measure of intellectual worth.
- Read O'Reilly book "MS Office Annoyances", check the web site annoyances.org
- Don't let PowerPoint lead you to the Gettysburg Address syndrome…

Summary

- Topics for a paper are everywhere; start with a question
- Writing the paper
  - Outline and major ideas first
  - Co-develop writing and testing/implementation simultaneously
  - Don't put off mechanics of formatting, bibliography until end
- Presenting results
  - Use different mechanisms than for paper: less formal
  - Present slightly different results, tell different story, but same conclusion as paper
  - It's a conversation more than a lecture. You can draw benefit beyond just publicity for yourself.