Armando's Paper Writing and Presentations Page

- Hints for Giving a Good SURF, REU, etc. Talk
- Hints for Technical Paper Writing
- Prof. John Ousterhout's Hints for Reviewing Papers
- How and How Not to Write a Good Systems Paper, by SOSP-9 chairs Roy Levin and Dave Redell
- How to write a good OOPSLA paper. The comments apply equally well to other conferences, IMHO.
- Related: Prof. Dave Patterson's *How to Have a Bad Career In Research/Academia*

Hints for Giving a Good SURF, REU, etc. Talk

Outline:

1. Know thy audience and watch thy jargon
2. Keep the big picture in mind
3. Tell me a story, don't read me an article
4. Pace yourself
5. What did you just say again?
6. There will be questions...

Know thy audience and watch thy jargon

Know your audience! At typical REU/SURF presentations, the audience consists of a wide variety of people from different disciplines. Roughly, the structure of your talk should reflect the following goals: (a) hook everyone (including those not in your area) on your topic/problem; (b) impress the experts with your specific work; (c) wrap up and recapture the attention of the non-experts. The fraction of your talk devoted to each of these depends on the level of sophistication of your audience.

Framing the problem can be a challenge. Are there popular-press or "real-world" examples you can appeal to in order to illustrate what you're doing? Even examples from Star Trek can be useful, since a lot of technology areas are rapidly approaching the abilities envisioned in that venerable show. Graphics, photos, and short video clips are also useful. Your project advisor can probably help you find something appropriate.

Beware of jargon, which covers both terms and concepts. Some people don't know what public-key cryptography is; others are specialists and just want to know which cryptosystem you used. Some people aren't familiar with the *concept* of separating a user interface from a device or application; plan to explain it if needed.

Keep the big picture in mind

What is the high level view of what's going on? You need to motivate the overall project to an audience who might be thoroughly unfamiliar with it. In a SURF talk, it might take up to 1/3 of the talk to motivate the problem and be sure everyone understands (at least at a high level) why it's useful, interesting, etc. This is time well spent: if people don't understand the ultimate goal, they probably won't pay attention to what you did.

How is the big picture divided into subproblems, and where do you fit in? Now that the big picture is clear, what are the specific subproblem challenges? Which part of which subproblem are you working on? Some of this will necessarily get into details that not everyone in the audience can follow.

If your efforts succeed, what will you have demonstrated? Another way to ask this is: what is the "research question" (or questions) being addressed here? In other words, ten years from now, when the hardware, software, etc. have all changed and the computers of the day make today's computers look like Tinkertoys, what *fundamental nugget of an idea* will still be considered relevant and applicable? This is often very hard to identify, and it may be that your own piece of the project contributes only a small part toward forming that Big Idea, but research is a building that has to be built one painful brick at a time. (Ask any Ph.D. student.)

Tell me a story, don't read me an article

Rehearse your talk enough that you don't need paper notes (or, at most, minimal notes - two or three 3x5 index cards for the *whole talk*). If you make eye contact, engage your audience, and tell them a story, they will pay attention. Try not to read from notes; they can read a paper as well as anyone. Having a speaker bring the material to life is what makes a talk different and potentially a better avenue for communicating your work to a lay audience.

Don't be afraid to use humor. If a funny picture, animation, joke, etc. is appropriate, it keeps the audience interested. But don't fall into the extremely annoying trap of using these gratuitously; it distracts the audience and gives the impression that you are using these to cover a lack of competence with the material.

Pace yourself

You won't have time to say everything you want. The higher level the talk, the less detail you'll have time for. A time-tested rule of thumb is: 2 minutes per slide. This sounds conservative, but it is very well borne out on average. Therefore, exlcuding the title/outline and conclusion slides, you should have half as many slides as you have minutes to speak.

Find a couple of key "timepoints" in the talk ("By the time I get to this slide, I should be *n* minutes into the talk"). Once again, rehearsal is key to debugging this. Remember, if you run out of material early, you are still prepared with a level of detail deeper than your talk, so you can use the extra time to elaborate on a particular point of interest to you; but if you are running short of time, you won't be able to communicate everything you want to say, and your audience will not come away with a representative picture of what you did or why they should care.

What did you just say again?

Especially if the middle part of your talk is aimed at technical experts, be sure you recap towards the end what the overall problem was and what your contribution was. Plan on 1-2 slides for this. People best remember the beginning and the end, so make sure these are rock solid. (Ask anyone who has written a Broadway musical if you don't believe this.) It may be appropriate to include "future work" here - things left to be done (some of which may have been discovered as a *result* of your work, which is always good) and new issues that came up as a result of your work.

There will be questions...

People will ask about stuff not in your talk. The main preparation/rehearsal, then, is to know your material at one level of detail *deeper* than your slides. Usually you cannot predict the questions; so, although you should make sure you can explain every point on your slides in additional detail if necessary, do not expect that those are the only questions you will get. People remember how well you handled your questions, since it demonstrates real familiarity with your material (anyone can rehearse and deliver a prepared talk on a topic they know little about).

Helpful Hints for Technical Paper Writing

Acknowledgments: Particularly influenced by Seth Hutchinson (MS thesis advisor), Eric Brewer (PhD thesis advisor), John Mullin (high school English--really!), and benefited from proofreadings by too many people to mention by name.

Note: This is a page about writing technical papers, but many of the techniques also seem applicable to both non-technical writing and giving presentations.

Vision, Implementation, and Survey Papers

In a vision paper, you describe your grand scheme of the world and why it is good. You need some data to back up your statements, but this is not a detailed measurements paper. The goal of this paper is to convince the reader that your scheme is interesting, different, better than other schemes that have addressed similar problems, raises legitimate research questions, and is therefore worth spending the time to pursue research on.

- If you're writing a vision paper, you have to be absolutely convinced of your vision, or no one else will be. Make no statement that cannot be backed up by citation, quantitative data, or at least a very good first-cut experiment ("preliminary results suggest....")

The implementation paper, by contrast, gives detailed measurements of a system that was perhaps described in a previous vision paper. The goal here is to demonstrate what you learned from actually building the system: Did it validate your research hypothesis? What came out differently than you expected, and why? How much better, quantitatively, is your design than others'?

- If you're writing a measurement-and-results paper, first determine which graphs will convey the results you think are important. Given those, the paper will practically write itself.

Survey papers: TBD...

Before You Write...

- If possible, present your work in a short 5-10 minute talk to your colleagues before starting to write. This helps identify strengths and weaknesses and will give you an idea of what other people see as the important contributions. Surprisingly often, they will spot a significant contribution that you totally overlooked, or suggest a novel application of your ideas that dramatically increases the relevance and impact of your paper.
- Don't cram. Recall the old saw about how nine pregnant women cannot produce a baby in one month. You can't throw all your time into a paper at the last minute and expect a good result: you will become saturated, lose perspective because you are too close to the material, and ultimately be spinning your wheels, changing stuff back and forth without a really good feeling for why you're doing it. Exceptions to this occur, but they're rare.
- Know when to say when. Even if you have written the paper with plenty of time and had a lot of outside review, after a certain point you will not be able to add much value without taking a break for a while (maybe a week or two). When that point comes, further work on the paper is just thrashing and not likely to improve it much, though it will leave you feeling dissatisfied. Wait for the reviews from the PC before doing much more.

Starting Checks

- Write from an outline. Let me say that again, because it's really important: write from an outline. I know no one who can reel off any coherent technical writing more than one page long without some kind of top-down strategy. At least sketch out the major sections of the paper, and what points you want to make in each, from 10000 feet. If you write any complete sentences during this phase, you're getting mired in detail already. Bullets are what you want.
- Don't even try to write the title or abstract until after the whole rest of the paper is written. Then, and only then, will you actually know what the hell it is you want to say.
- Unless you're writing a PhD thesis, your paper will make only a small number of discrete points--say 2 to 4. Each important point should appear 3 times: once in the abstract/introduction, once in the body of the paper (where it is explained in detail), and once in the conclusions (where you derive some implications of this point for the future of systems research, or whatever). Bulleted conclusions can help. Remember that conference referees are at least as busy as you and they have to read several of these. Make sure they remember yours.

The Actual Paper: Writing

1. start from the outline.
2. Make the outline reflect the level of subsections: for each subsection, write *no more than two lines* describing the purpose/goal of that subsection. This text will **NOT** be part of the paper - it is only there to remind you what you are trying to accomplish. It is **ESSENTIAL** that you be able to capture the purpose of a subsection in one or two lines. If you cannot do this, then you probably don't understand what the subsection is really about, and when you try to write the text, it will be jumbled.
3. Then, for each subsection, map out specific paragraphs: for each paragraph, write *one* sentence that explains the topic or main goal of *just that paragraph*. Again, this sentence probably will NOT make it into the actual text. It's important to keep it to *one* sentence. (As every style manual will tell you, including Strunk & White, virtually every well-formed paragraph does indeed have *one* sentence that explains the point of the paragraph, with the other sentences supporting or expanding on the point of the topic sentence.) If you cannot fit the point of the pargraph into 1 sentence, the paragraph is probably making >1 point, so should be split into multiple paragraphs.
4. Read through everything you have written and see if it has a logical flow, ie if you believe it represents your work adequately.
5. Give what you have written to a technical colleague *completely unfamiliar* with your work

(but able to understand the computer science part), have them read it, then have them tell you (without looking at it) what s/he thinks the main point and contributions are.

6. If all goes well, *now* replace the topic sentences with complete paragraphs.

   This way of writing will not yield a shakespearean work of literature, but it is consistent and will result in readable, logically organized prose by construction.

The Actual Paper: Revising/Editing

- Your section organization will change. Sometimes it will be shuffled dramatically. This is fine; it means you're understanding what presentation order works best. If you don't go through at least three or four *major* revisions (where you move around or chop entire sections), it's probably lousy.
- After doing some edits on each draft, give it a full top-to-bottom reading to evaluate its coherence and flow of ideas. Then, take a couple of hours and do something else; once you get close enough to your paper, you start missing the forest for the trees.
- Even early drafts are valuable for getting your colleagues' comments. Get comments from people who you think may be skeptical of your approach. Get comments from people who will really rip your writing style apart. Remember, at least they are your friends; the conference referees probably are not.
- Cite, cite, cite! Ask your colleagues for suggestions and pointers. You never want to be asked: "What about the work done by xxx, which obviously has something in common with your own?" (or worse: "...which refutes your own?") Give due credit to those whose efforts you build on, as well as pointing out how your approach is different from (and better than) previous ones.

About Writing

It's often said, correctly I think, that most technical people don't write well. This doesn't mean that they lack knowledge of grammar or spelling (though this is sometimes the case), but that they don't know how to organize their writing at the level of paragraphs.

- Don't artificially formalize your writing style. Technical writing must be clear and concise. Overblown writing rarely fools anyone and it makes the paper boring to read.
  Bad: "Problem X is clearly a critical area that impacts our research agenda and hypothesis. Our ideas about problem X are embryonic and still evolving, and doubtless our ongoing work in this area will quickly yield fruitful results."
  Better: "We recognize that problem X is central to our agenda, but we have only begun to investigate it."
- If you haven't read Strunk and White's *The Elements of Style*, read it now. If you have, read it again. If you can't organize a paragraph, you won't have much luck organizing a chapter.
- Omit needless words. Don't be surprised if this turns out to be 30-40% of the words you originally wrote. Your first effort rarely captures the most vigorous or concise way to say something. Spend time tersifying.
- Run your paper by someone who is anal retentive about grammar to catch common errors: misuse of *which* and *that*, non-words and non-phrases such as *for all intensive purposes* or *irregardless,* lack of parallel sentence structure...

Final Checks

Remember that this will be read by people who (a) have never heard of you and the review is anonymous anyway, (b) have never heard of your project, (c) are reading about 15-20 papers apiece, all in different subject areas. They will spend the first 5 minutes deciding if your paper is actually good enough to be worth a fully detailed read; they will then spedn an hour or so reading it in detail, trying to figure out (a) what your contribution is, (b) if the contribution is substantial enough to be worth publishing, (c) if the contribution is "feasible" (ie it is implementable and therefore would be useful to someone).

- Does the paper make clear *precisely* what *your new* contributions are, and how they are different/better than existing approaches to this or similar problems?
- Does the outline of the paper (sections, subsections, etc.) cohere regardless of the granularity at which you view it? (The Outline mode of MS Word is a valuable feature for this check. I also wrote a simple Perl script that does this for LaTeX files.)
- Have you observed the following invariant: Before telling me what you did, tell me why I should care.
- Have you made every important point three times--once in the introduction/abstract, once in the body of the paper, and once in the conclusions? (Bulleted conclusions are usually a good idea)
- Have you had it read by at least one person familiar with each of the areas the paper impinges on? (Think of them as consultants in that area. There is a risk that you will get some of the details wrong in talking about an area that is tangential to the paper but that you're not very familiar with, and if a reviewer happens to be versed in that area, it decreases your credibility. Such references are easy to get right, so there is no excuse.)
- Have you searched carefully for any related work, and properly acknowledged it? The availability of papers and search indices on the Web makes it worse than ever to overlook significant related work.
- Are you able to capture the non-experts in the audience with the opening of your paper, and impress the experts in the body of the paper?
- Can you read only the abstract and conclusions and be able to give someone else a 30-second digest of what the paper claims it says?

John Ousterhout's Hints for Reviewing Papers

I preface these with some high-level questions of my own that I try to answer quickly on a first pass. Note that the answer to each question tells you something about the technical content of the paper, whereas the *ease of extracting the answer* to each question tells you something about the quality of the writing. For example, a paper may have a really great main contribution that is so poorly expressed that it takes you a couple of passes just to figure out what the paper is "really" about.

- Is this a vision/position/direction paper, or a measurement/implementation paper?
- If I know the area well, can I mentally slot this paper somewhere in the taxonomy? ("Differs from X as follows; has the following in common with Y;" etc.) If the paper is radically brilliant, new, or iconoclastic work, this question may not apply.
- Can I summarize the single most important contribution in one or two sentences?

John Ousterhout delivered the following wisdom to his UC Berkeley CS 262 (advanced topics in operating systems) class in Fall 93, as the ISCA deadline approached.

Issues

- Will this advance the state of the art?

- Did you learn anything new?
- Does it provide evidence which supports/contradicts hypotheses?
- Experimental validation?
- Will the paper generate discussion at the conference?
- How readable is the paper? (The draft can be modified, and if the ideas are very important, you may accept it anyway.)
- Is the paper relevant to a broader community?

Goals of Review

- Guide the program committee in selection process
- Help authors (to revise paper for acceptance, to understand rejection, to improve further research and future projects)

Structure

- 1/2 to 1 page of text (2 - 4 paragraphs)
  Longer reviews are generally given for better papers, shorter reviews for bad papers
- 1 paragraph executive summary
  - what is the paper trying to do?
  - what is potential contribution of paper?
  - short summary of strengths and weaknesses (sentence or 2)
  - accept/reject (hard, because you don't necessarily see the entire sample)
- several paragraphs of details (listed in order of importance)
  - technical flaws?
  - structure of paper?
  - are key ideas brought out?
  - don't want to just describe system, also need motivation and justification of approach
  - presentation? (ex: undefined terms, confusing wording, unclear sections...)
  - justification -- can they say why ideas are important?
  - comparison with other systems? For bigger conferences (SOSP, ISCA, ASPLOS) need quantitative evidence of ideas
  - grammar? (usually only point out consistent errors)

Short rules to triple-check your paper

- Pat Hanrahan says: "Future Work section must be earned". If you haven't made us care about your contribution thus far, we won't care to read Future Work either.
- THere's 3 kinds of statements in a systems paper: statements supported by citations, statements supported by experiments, and opinions. Avoid opinions.
- Good reviewers are overloaded. Remember, Butler Lampson is looking for an excuse to stop reading your paper. Don't give him one.

---

fox@cs.stanford.edu