

Discussion of last week's homework

- Which papers did you select? How?
- How did you identify the seminal paper?
- Summarize the paper you've reviewed
 - Contributions?
 - Technical merit?
 - Writing?

Writing research papers

Writing papers is a skill

- Many papers are badly written
- Good writing is a skill you can learn
- It's a skill that is worth learning:
 - You will get more brownie points (more papers accepted, etc.)
 - Your ideas will have more impact
 - You will have better ideas

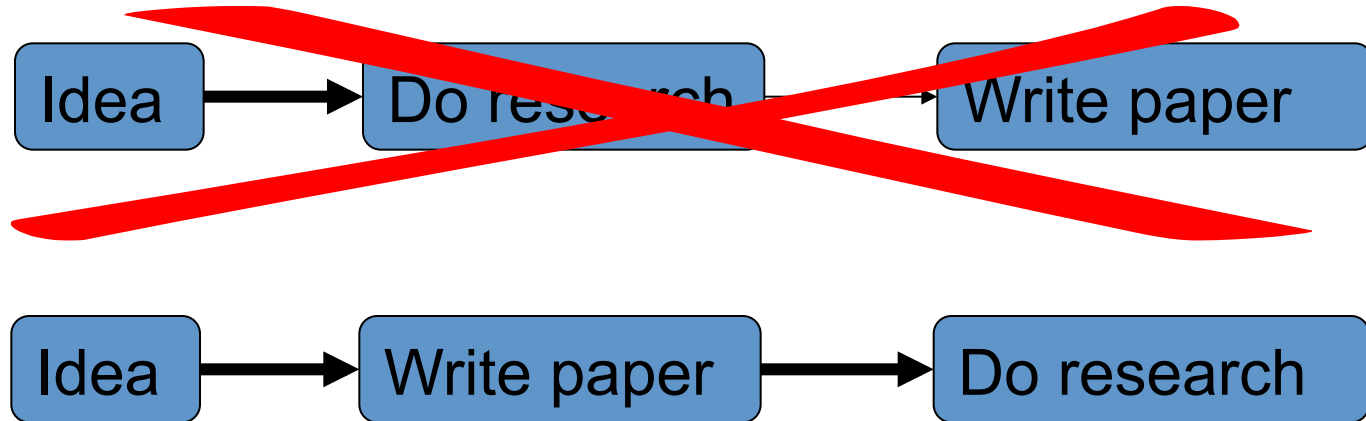
Increasing importance



Writing papers: model 1



Writing papers: model 2



- Forces us to be clear, focused
- Crystallises what we don't understand
- Opens the way to dialogue with others: reality check, critique, and collaboration

Do not be intimidated

Fallacy

You need to have a fantastic idea before you can write a paper. (Everyone else seems to.)

Write a paper,
and give a talk, about

any idea,

no matter how weedy and insignificant it may
seem to you

Do not be intimidated

Write a paper, and give a talk, about any idea,
no matter how insignificant it may seem to
you

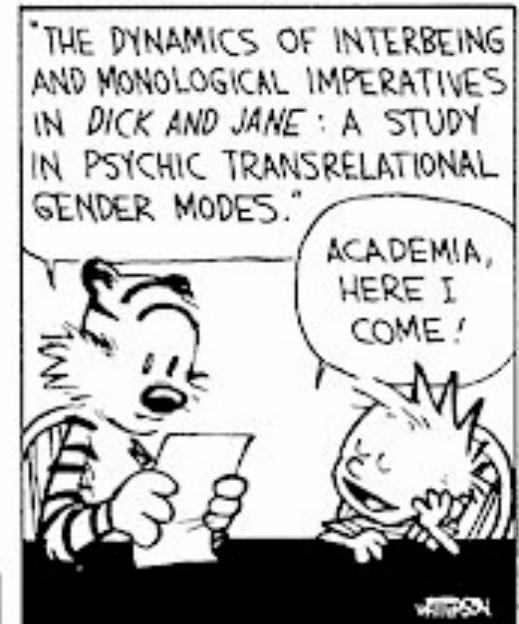
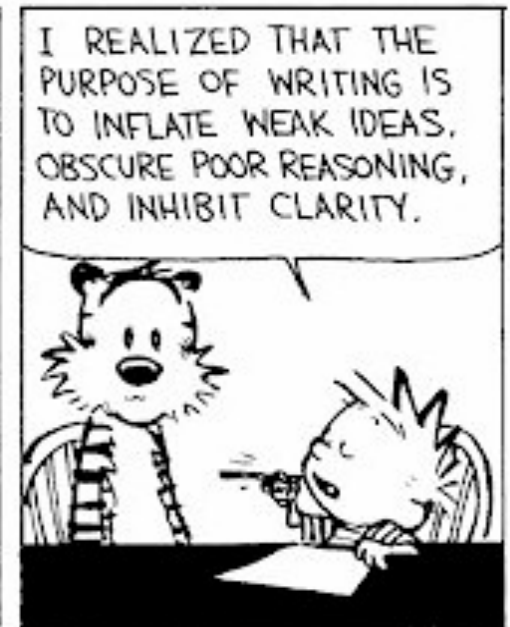
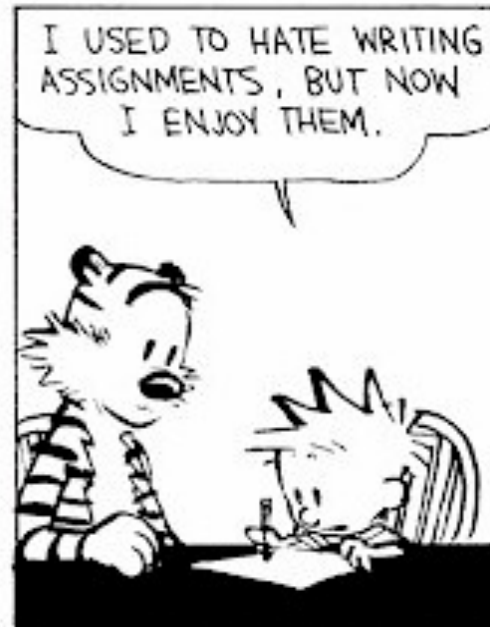
- **Writing the paper is how you develop the idea in the first place**
- It usually turns out to be more interesting and challenging than it seemed at first

THE PURPOSE OF YOUR PAPER

Why bother?

Fallacy

we write papers and give talks mainly to impress others, gain recognition, and get promoted



Why bother?

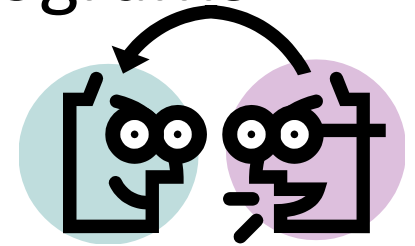
Fallacy

we write papers and give talks mainly to impress others, gain recognition, and get promoted

Good papers and talks are a fundamental part of research excellence

Papers communicate ideas

- Your goal: to infect the mind of your reader with **your idea**, like a virus
- Papers are far more durable than programs (think Mozart)



The greatest ideas are (literally)
worthless if you keep them to yourself

The Idea

Idea

A re-usable insight,
useful to the reader

- Figure out what your idea is
- Make certain that the reader is in no doubt what the idea is. Be 100% explicit:
 - “The main idea of this paper is....”
 - “In this section we present the main contributions of the paper.”
- Many papers contain good ideas, but do not distil what they are.

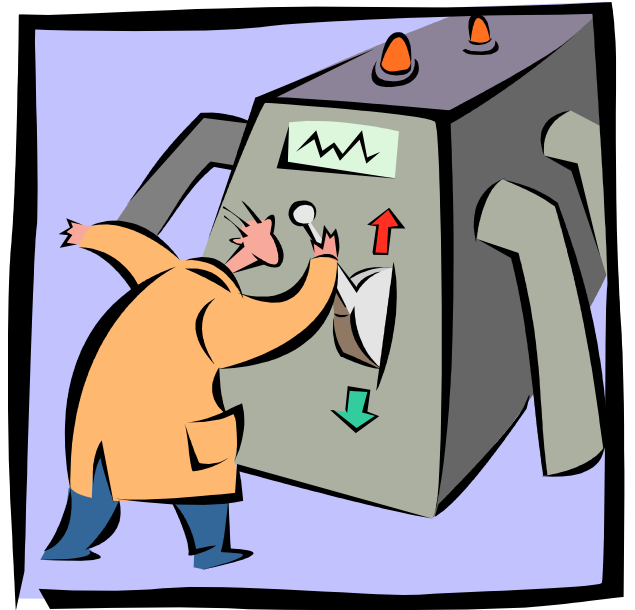
One ping

- Your paper should have just one “ping”: one clear, sharp idea
- Read your paper again: can you hear the “ping”?
- You may not know exactly what the ping is when you start writing; but you must know when you finish
- If you have lots of ideas, write lots of papers

Thanks to Joe Touch for “one ping”

The purpose of your paper is not...

To describe the
WizWoz system



- Your reader does not have a WizWoz
- She is primarily interested in re-usable brain-stuff, not executable artefacts

Golden Rule: Storytelling

- Every paper tells a story
 - Not the chronology of your research
- What's the big deal? The main idea?
 - What is the problem?
 - Why is it hard?
 - Why is your solution interesting, significant?
 - Why should the reader care?
- Note: Your story is not a mystery novel.
 - Write top-down!
- Note: Nobody is as interested in this topic as you
 - Make it interesting!

Papers are not Novels!

- Many aspects of writing style that are appropriate in literature are ineffective (or annoying!) in technical papers
- Tips for simplicity
 - One idea/topic per logical unit
 - Simple organization
 - Short words, simple structure

Economize



- "I am sorry I have had to write you such a long letter, but I did not have time to write you a short one" -- Blaise Pascal
- The length of a paper should be correlated with its content

THE STRUCTURE OF YOUR PAPER

Your narrative flow

- Here is a problem
- It's an interesting problem
- It's an unsolved problem
- **Here is my idea**
- My idea works (details, data)
- Here's how my idea compares to other people's approaches

I wish I knew how to solve that!

I see how that works. Ingenious!



Structure (conference paper)

- Title (1000 readers)
- Abstract (4 sentences, 100 readers)
- Introduction (1 page, 100 readers)
- The problem (1 page, 10 readers)
- My idea (2 pages, 10 readers)
- The details (5 pages, 3 readers)
- Related work (1-2 pages, 10 readers)
- Conclusions and further work (0.5 pages)

The abstract

- I usually write the abstract last
- Used by program committee members to decide which papers to read
- Four sentences [Kent Beck]
 1. State the problem
 2. Say why it's an interesting problem
 3. Say what your solution achieves
 4. Say what follows from your solution

Example

1. Many papers are badly written and hard to understand
2. This is a pity, because their good ideas may go unappreciated
3. Following simple guidelines can dramatically improve the quality of your papers
4. Your work will be used more, and the feedback you get from others will in turn improve your research

Structure

- Abstract (4 sentences)
- **Introduction** (1 page)
- The problem (1 page)
- My idea (2 pages)
- The details (5 pages)
- Related work (1-2 pages)
- Conclusions and further work (0.5 pages)

Introduction

- Summarizes the whole story
 - The most important part of the paper!
 - If people don't understand your problem, approach, importance by the end of the intro, you're out of luck
- Two schools of thought
 - Write it first: make certain the story is clear
 - Write it last: story becomes clear at the end
- My advice: do both

Why to Start the Intro Early

- It's important to be able to concisely summarize your key contributions
 - In as little as a single paragraph
- If you cannot do this, it's quite possible that your thinking is not clear
 - Working on the story can improve your thinking
 - It may also become clear that you don't have a paper!
- **Muddled writing reflects muddled thinking**

The Importance of First Impressions

- Many readers make up their minds within the first few paragraphs
- The first few paragraphs should state the paper's purpose *with context*
 - Beware “This paper concerns”
- The beginning should be intelligible to *any* reader

Introduction: J. Kurose's Formula

- Paragraph 1: Context
- Paragraph 2: Problem area
- Paragraph 3: "This paper ..."
- Paragraphs 4-5: Challenges / Solutions
- Paragraph 6: Summary of results
- Paragraph 7: Outline

The introduction (1 page)

- 1. Describe the problem**
- 2. State your contributions**

...and that is all

ONE PAGE!

Describe the problem

1 Introduction

There are two basic ways to implement function application in a higher-order language, when the function is unknown: the *push/enter* model or the *eval/apply* model [11]. To illustrate the difference, consider the higher-order function **zipWith**, which zips together two lists, using a function **k** to combine corresponding list elements:

```
zipWith :: (a->b->c) -> [a] -> [b] -> [c]
zipWith k []      []      = []
zipWith k (x:xs) (y:ys) = k x y : zipWith xs ys
```

Here **k** is an *unknown function*, passed as an argument; global flow analysis aside, the compiler does not know what function **k** is bound to. How should the compiler deal with the call **k x y** in the body of **zipWith**? It can't blithely apply **k** to two arguments, because **k** might in reality take just one argument and compute for a while before returning a function that consumes the next argument; or **k** might take three arguments, so that the result of the **zipWith** is a list of functions.

Use an example to introduce the problem

State your contributions

- Write the list of contributions first
- The list of contributions drives the entire paper: the paper substantiates the claims you have made
- Reader thinks “gosh, if they can really deliver this, that’s be exciting; I’d better read on”

State your contributions

Which of the two is best in practice? The trouble is that the evaluation model has a pervasive effect on the implementation, so it is too much work to implement both and pick the best. Historically, compilers for strict languages (using call-by-value) have tended to use `eval/apply`, while those for lazy languages (using call-by-need) have often used `push/enter`, but this is 90% historical accident — either approach will work in both settings. In practice, implementors choose one of the two approaches based on a qualitative assessment of the trade-offs. In this paper we put the choice on a firmer basis:

- We explain precisely what the two models are, in a common notational framework (Section 4). Surprisingly, this has not been done before.
- The choice of evaluation model affects many other design choices in subtle but pervasive ways. We identify and discuss these effects in Sections 5 and 6, and contrast them in Section 7. There are lots of nitty-gritty details here, for which we make no apology — they were far from obvious to us, and articulating these details is one of our main contributions.

In terms of its impact on compiler and run-time system complexity, `eval/apply` seems decisively superior, principally because `push/enter` requires a stack like no other: stack-walking

Bulleted list of contributions

Do not leave the reader to guess what your contributions are!

Contributions should be irrefutable

NO!	YES!
We describe the WizWoz system. It is really cool.	We give the syntax and semantics of a language that supports concurrent processes (Section 3). Its innovative features are...
We study its properties	We prove that the type system is sound, and that type checking is decidable (Section 4)
We have used WizWoz in practice	We have built a GUI toolkit in WizWoz, and used it to implement a text editor (Section 5). The result is half the length of the Java version.

No “rest of this paper is...”

- Not:

“The rest of this paper is structured as follows. Section 2 introduces the problem. Section 3 ... Finally, Section 8 concludes”.
- Instead, **use forward references from the narrative in the introduction.**

The introduction (including the contributions) should survey the whole paper, and therefore forward reference every important part.

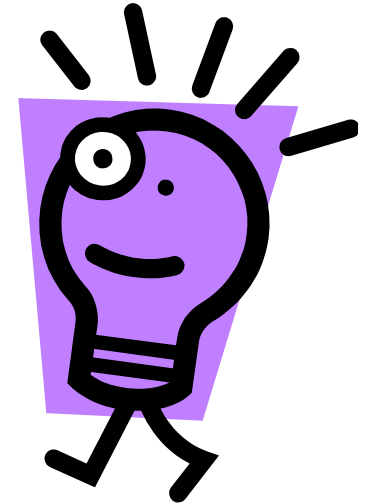
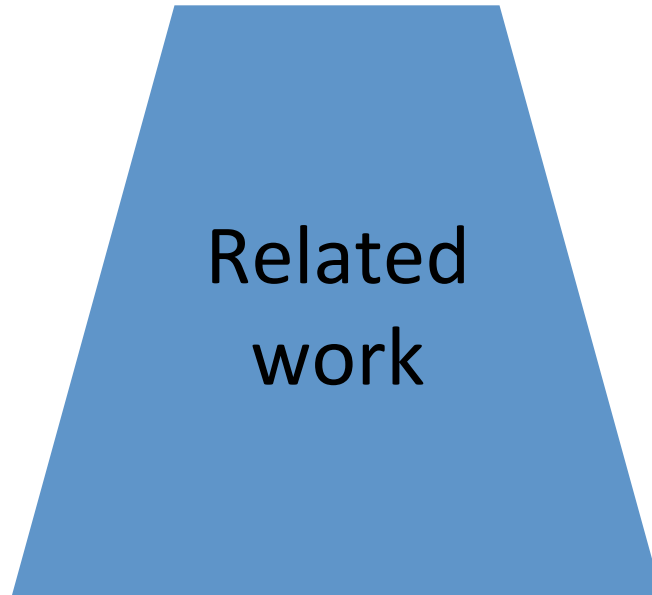
Structure

- Abstract (4 sentences)
- Introduction (1 page)
- ~~Related work~~
- The problem (1 page)
- My idea (2 pages)
- The details (5 pages)
- Related work (1-2 pages)
- Conclusions and further work (0.5 pages)

No related work yet!



Your reader



Your idea

We adopt the notion of transaction from Brown [1], as modified for distributed systems by White [2], using the four-phase interpolation algorithm of Green [3]. Our work differs from White in our advanced revocation protocol, which deals with the case of priority inversion as described by Yellow [4].

No related work yet

- **Problem 1:** the reader knows nothing about the problem yet; so your (carefully trimmed) description of various technical tradeoffs is absolutely incomprehensible
- **Problem 2:** describing alternative approaches gets between the reader and your idea



(Un)Related Work Section

- Section 2, or Penultimate Section?
 - Placing early pushes the “meat” of the paper later, but can prevent the reader from discounting your technique
- Handwavy rule
 - Generally better to put towards the end, but...
 - If the topic of the paper appears similar to others, have an “unrelated work” section after the intro

Structure

- Abstract (4 sentences)
- Introduction (1 page)
- The problem (1 page)
- My idea (2 pages)
- The details (5 pages)
- Related work (1-2 pages)
- Conclusions and further work (0.5 pages)

Presenting the idea

3. The idea

Consider a bifurcated semi-lattice D , over a hyper-modulated signature S . Suppose p_i is an element of D . Then we know for every such p_i there is an epimodulus j , such that $p_j < p_i$.

- Sounds impressive...but
- Sends readers to sleep
- In a paper you **MUST** provide the details, but **FIRST** convey the idea

Presenting the idea

- Explain it as if you were speaking to someone using a whiteboard
- **Conveying the intuition is primary**, not secondary
- Once your reader has the intuition, she can follow the details (but not vice versa)
- Even if she skips the details, she still takes away something valuable

Putting the reader first

- **Do not** recapitulate your personal journey of discovery. This route may be soaked with your blood, but that is not interesting to the reader.
- Instead, choose the most direct route to the idea.

The payload of your paper

Introduce the problem, and your
idea, using

EXAMPLES

and only then present the
general case

Using examples

2 Background

To set the scene for this paper, we begin with a brief overview of the *Scrap your boilerplate* approach to generic programming. Suppose that we want to write a function that computes the size of an arbitrary data structure. The basic algorithm is “for each node, add the sizes of the children, and add 1 for the node itself”. Here is the entire code for `gsize`:

```
gsize :: Data a => a -> Int
gsize t = 1 + sum (gmapQ gsize t)
```

The type for `gsize` says that it works over any type `a`, provided `a` is a *data* type — that is, that it is an instance of the class `Data`¹. The definition of `gsize` refers to the operation `gmapQ`, which is a method of the `Data` class:

```
class Typeable a => Data a where
  ...other methods of class Data...
  gmapQ :: (forall b. Data b => b -> r) -> a -> [r]
```

Example
right
away

The details: evidence

- Your introduction makes claims
- The body of the paper provides **evidence to support each claim**
- Check each claim in the introduction, identify the evidence, and forward-reference it from the claim
- Evidence can be: analysis and comparison, theorems, measurements, case studies

State the results carefully

- Clearly state assumptions
- Experiment/simulation description
 - Enough info to nearly recreate experiment
- Simulation/measurements:
 - Statistical properties of your results
 - E.g., confidence intervals
- Are results presented representative?
 - Or just a corner case that makes the point you want to make

Don't overstate/understate your results

- Overstatement mistake:

- “We show that X is prevalent in the Internet”
- “We show that X is better than Y”

when only actually shown for one/small/limited cases

- Understatement mistake: fail to consider broader implications of your work

- If your result is small, interest will be small
- “rock the world”

Evaluation Section

- **Context:** Clearly state assumptions
 - In what context do your results hold?
 - How general are they?
- **Recipe:** Clearly describe the setup
 - Machines, data, scripts, topologies, etc.
 - You must make this clear!
 - Rule of thumb: The reader should be able to recreate the experiment and results from the description in the paper
- We will have a full lecture later on analysis (and presentation) of results

Evaluation Section

- Many people will skim
 - Corollary: Make it skimmable!
- Evaluation signposts
 - Table summarizing key results (and where to find them in the paper)
 - Declarative subsection headings
 - Trick: Finding as subsection heading
 - Readable graphs
 - Captions that summarize the key finding
 - (implication: each graph should have **one** main point)
 - Big fonts!

Structure

- Abstract (4 sentences)
- Introduction (1 page)
- The problem (1 page)
- My idea (2 pages)
- The details (5 pages)
- **Related work** (1-2 pages)
- Conclusions and further work (0.5 pages)

Related work

Fallacy

To make my work look good, I have to make other people's work look bad

The truth: credit is not like money

Giving credit to others does not diminish the credit you get from your paper

- Warmly acknowledge people who have helped you
- Be generous to the competition. “In his inspiring paper [Foo98] Foogle shows.... We develop his foundation in the following ways...”
- Acknowledge weaknesses in your approach

Credit is not like money

Failing to give credit to others can
kill your paper

If you imply that an idea is yours, and the referee knows it is not, then either

- You don't know that it's an old idea (bad)
- You do know, but are pretending it's yours (very bad)

Structure

- Abstract (4 sentences)
- Introduction (1 page)
- The problem (1 page)
- My idea (2 pages)
- The details (5 pages)
- Related work (1-2 pages)
- Conclusions and further work (0.5 pages)

Conclusion Section

- Keep it crisp
- Remember how reviewers and readers skim papers (intro, abstract, conclusion)
- Two elements
 - Very concise summary (one paragraph)
 - Remember, readers by now should have context
 - “Elevation” (one paragraph to one page, depending on the paper)
 - What are the takeaways? General lessons or applications?
 - Broader implications?

STYLE AND COMPOSITION

Style and Composition

- Often referred to as “flow”
 - How sentences flow together to form paragraphs
 - How paragraphs flow together to form sections
 - How sections flow together to form a paper
- The most important aspect of writing a paper

Organizing Paper Flow

- **Plan first, write later**
- Write top-down
 - Step 1: Outline sections
 - Step 2: Within a section, outline paragraphs
 - For each paragraph, write topic sentences

Writing a Section, Top-Down Style

- Make a bulleted list of points to include
- Cluster the points into related topics/points
- For each cluster, write a topic sentence
- Organize your topic sentences
- Make subsections if necessary
- Fill in paragraph details (top down!)
- Add paragraph headings

Style Points

- Motivation
 - Everything that a paper includes should be accompanied with an explanation for why it is necessary/interesting useful
- Balance
 - Topics of equal relevance should be addressed with equal weight/length

Writing a Paragraph

- A paragraph is group of **logically related** sentences
- Start with a sentence that describes the logical relationship (“thread”)
- Keep continuity
 - Keep a common verb tense
 - Don’t string together loosely related sentences

Signposting

- The reader must have a clear view of how the paper/story will proceed
- Allow for *top-down reading*
- **Signposts:** How is the paper (or section) organized?
 - Outline at end of the introduction
 - Preamble to each section
 - Declarative subsection titles
 - Paragraph headings

Landscaping

- Your goal: efficient information transfer
 - Forcing the reader to “block” or “context switch” by taking a break, falling asleep, or, worse---skimming over important points---defeats the purpose
- Consecutive pages of dense text: ouch!
 - Tables
 - Figures
 - Whitespace
 - Signposts

Visual structure

- Give strong visual structure to your paper using
 - sections and sub-sections
 - bullets
 - italics
 - laid-out code
- Find out how to draw pictures, and use them

THE PROCESS OF WRITING

The writing process

- Start early. Very early.
 - Hastily-written papers get rejected.
 - Papers are like wine: they need time to mature
- Collaborate
- Use SVN to support collaboration

The writing process

- Prepare first, then write
 - Take time to crystallize your thoughts
 - Clear thoughts lead to clear writing
 - Much more difficult to revise muddled text...often you will start over!
- Shut off all distractions
 - Writing takes focused, clear thinking
 - Context switches and interrupts are particularly damaging

The Writing Process: Growth

- Practice whenever possible
 - Write a lot, multiple times per day if possible
 - Email, notes, blogs, publications
- Find a style that you like and try to emulate it
- Experiment

Editing: Reading

- Read aloud
 - Helps identify clunky, awkward, and repetitive passages
- Read in reverse
 - Helps bypass your brain's tendency to fill in gaps, mistakes, etc.

Editing: Cutting

- Watch out for fancy words and cut them
- Toss out redundancy
- Each sentence, word, phrase, section, graph, etc. must be justified!
- Sleep on it...

Getting help

Get your paper read by as many friendly guinea pigs as possible

- Experts are good
- Non-experts are also very good
- Each reader can only read your paper for the first time once! So use them carefully
- Explain carefully what you want (“I got lost here” is much more important than “Jarva is mis-spelt”.)

Getting expert help

- A good plan: when you think you are done, send the draft to the competition saying “could you help me ensure that I describe your work fairly?”.
- Often they will respond with helpful critique (they are interested in the area)
- They are likely to be your referees anyway, so getting their comments or criticism up front is Jolly Good.

Listening to your reviewers

Treat every review like gold dust

Be (truly) grateful for criticism as well as
praise

This is **really, really, really** hard

But it's
really, really, really, really, really, really, really, really,
really, really, really
important

Listening to your reviewers

- Read every criticism as a positive suggestion for something you could explain more clearly
- DO NOT respond “you stupid person, I meant X”. Fix the paper so that X is apparent even to the stupidest reader.
- Thank them warmly. They have given up their time for you.

USAGE

Usage: Composing Individual Sentences

- Errors (spelling, grammar, etc.) or deviations in style can cause the reader to “context switch”
 - This creates a barrier for information flow
 - Your goal is to reduce or eliminate these
- Write in a style the reader expects
 - Reading previous conference proceedings can help here

Simplify Your Usage

- Never use a metaphor, simile, or other figure of speech which you are used to seeing in print
- Never use a long word where a short one will do
- If it is possible to cut a word out, always cut it out
- Never use the passive where you can use the active
- Never use a foreign phrase, a scientific word, or a jargon word if you can think of an everyday English equivalent
- Break any of these rules sooner than say anything outright barbarous

Use the active voice

The passive voice is “respectable” but it DEADENS your paper.
Avoid it at all costs.

NO

It can be seen that...

34 tests were run

These properties were
thought desirable

It might be thought that
this would be a type error

YES

We can see that...

We ran 34 tests

We wanted to retain these
properties

You might think this would
be a type error

“We” = you
and the
reader

“We” = the
authors

“You” = the
reader

Use simple, direct language

NO

The object under study was displaced horizontally

On an annual basis

Endeavour to ascertain

It could be considered that the speed of storage reclamation left something to be desired

YES

The ball moved sideways

Yearly

Find out

The garbage collector was really slow

Omit Needless Words!

- In order to => To
- The problem of optimizing => Optimizing
- the question as to whether => whether
- for optimization purposes => to optimize
- This is a module that => This module
- In a shorter running time => more quickly
- this is a subject that => this subject
- His story is a strange one. =>
His story is strange.

Avoiding Padding

- Adding together => adding
- Totally eliminated => eliminated
- Separated out => separated
- Give a description of => describe
- “the fact that” => ...
- “it is important to note” => ...

Writing Bugs

- Citations as nouns
 - “In [10], the authors showed ...”
 - Problem: forces the reader to context switch
 - Better: “Gray *et al.* previously showed ... [10].”
- Beginning a sentence with “However”
 - Problem: Not a qualified word
 - Better: “Unfortunately”, etc.

Writing Bugs

- The naked “This”
 - Problem: “this” is a modifier
 - “Next, we sample every tenth data point. This reduces processing time.”
 - Better: “Sampling every tenth data point reduces processing time.”
- Passive voice
 - “A request for content is sent to the server.”
 - *Who/what* performs the action?
 - Very important when specifying protocols, experimental setups, etc.

Which vs. That

- “Which” clauses can be removed from the sentence without changing the meaning
 - “BGP, which is the Internet’s routing protocol, ...”
 - They are always offset by commas
 - Better: omit “which is” entirely
- “That” clauses make the modified noun more specific and cannot be removed without changing meaning
 - “Can you send me the code that performs PCA on BGP routing updates?”
 - Not offset by commas

Other Misused Words

- Less vs. fewer
- Affect vs. effect
- Impact vs. influence
- May vs. can
- Further vs. farther
- Comprise vs. compose

The Articles (“a”, “the”, etc.)

- “A” / “an”
 - Non-specific modifier
 - “I need to work on a paper.” (implication: *any* paper)
- “The”
 - Specific modifier
 - “I need to write *the* paper.” (implication: specific paper)
 - “I need to read the papers.” (specific papers)
- Collective nouns often do not take any article
 - “Papers can provide useful background information.”
 - “The papers at *SIGCOMM* are very interesting this year.”

PUBLISHING

Basic stuff

- Submit by the deadline
- Keep to the length restrictions
 - Do not narrow the margins
 - Do not use 6pt font
 - On occasion, supply supporting evidence (e.g. experimental data, or a written-out proof) in an appendix
- Always use a spell checker

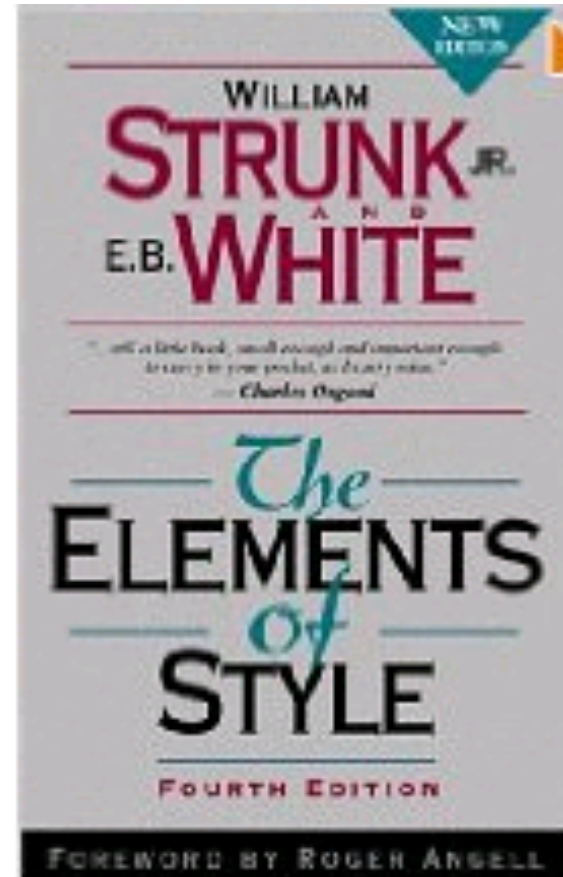
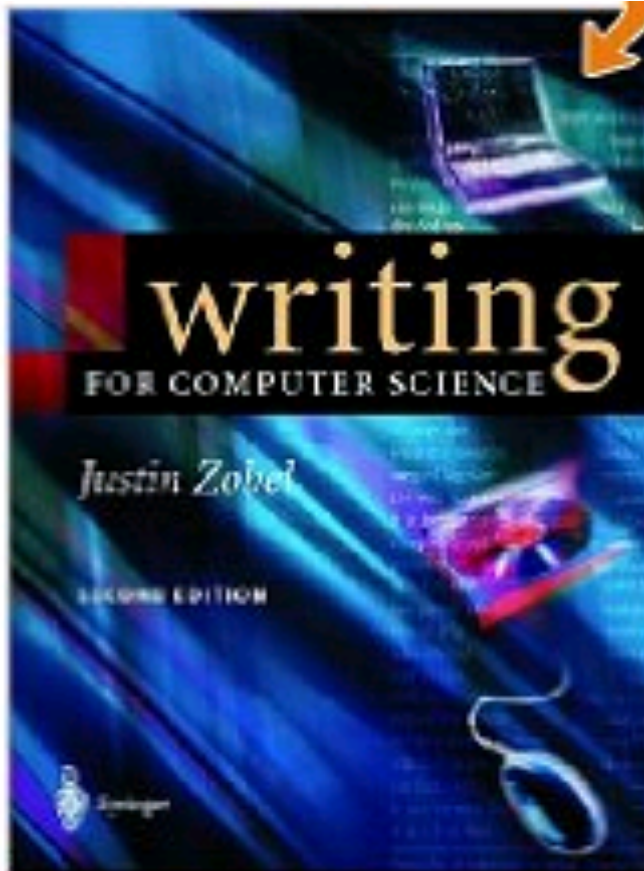
Publishing

- Publish where you will get mindshare, impact
- There *is life beyond* sigcomm, infocom
- Quality over quantity: avoid LPUs
- PhD dissertation != magnum opus
- Don't be driven by conference deadlines

Homework 3

- Start the writing of the project report
 - Introduction (the story of your paper)
 - Outline of the rest of the paper
 - Division into sections
 - Small description of what goes into sections
 - Related work
- Submissions in PDF
 - <https://tibre.lip6.fr/hotcrp/metho13-3/index>
 - Deadline: 8 Oct 2013 11pm CEST
 - Format on class web page
 - Write using latex
 - Limit = 2 pages

Recommended reading



Useful links

- J. Kurose, “Writing tips”
 - <http://www-net.cs.umass.edu/kurose/writing/>
- H. Schulzrinne, “Writing Technical Articles”
 - <http://www1.cs.columbia.edu/~hgs/etc/writing-style.html>
- S. Peyton-Jones, “Research skills”
 - <http://research.microsoft.com/en-us/um/people/simonpj/Papers/giving-a-talk/giving-a-talk.htm>
- S. Keshav, “Research advices”
 - <http://blizzard.cs.uwaterloo.ca/keshav/wiki/index.php/Aphorisms>
 - <http://ccr.sigcomm.org/online/files/p3-1v40n3a-keshav-editorial.pdf>