How to Write a Research Paper

Nicola Dragoni  
ndra@imm.dtu.dk  
Embedded Systems Engineering Section  
DTU Informatics  
Technical University of Denmark

Slides based on a talk by Simon Peyton Jones, Microsoft Research, Cambridge
Why bother?

1st Fallacy: we write papers and give talks mainly to impress others, gain recognition, and get promoted.
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

The greatest ideas are (literally) worthless if you keep them to yourself!
Do Not Be Intimidated

2nd Fallacy: you need to have a fantastic idea before you can write a paper or give a talk.

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

Writing the paper is how you develop the idea in the first place!
The Purpose Of Your Paper
The Purpose Of Your Paper Is...

To convey your idea!

... from your head to your reader’s head

Everything serves this single goal!
The Purpose Of Your Paper Is NOT...

To describe the WizWoz system

- Your reader does not have a WizWoz
- He is primarily interested in re-usable brain-stuff, not executable artefacts
Conveying the Idea

☑ Here is a problem

☑ It’s an interesting problem

☑ It’s an unsolved problem

☑ Here is my idea

☑ My idea works (details, data, prototype, ...)

☑ Here’s how my idea compares to other approaches
Follow Simple Guidelines…

☑ Many papers are badly written and hard to understand

☑ This is a pity, because their good ideas may go unappreciated

☑ Following simple guidelines can dramatically improve the quality of your papers

☑ Your work will be used more, and the feedback you get from others will in turn improve your research
The Structure of Your Paper

Abstract?

Introduction?

Related Work?

Conclusion?

Theorems?

Contribution?

Implementation?

Case study?

Figures?
Structure

- Abstract (~ 4 sentences)
- Introduction and contribution (~ 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)
The Abstract

I usually write the abstract last

Used by program committee members to decide which papers to read

Usually 4 “sentences”:

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

Security-by-Contract for Applications’ Evolution in Multi-Application Smart Cards

Nicola Dragoni\textsuperscript{1} and Olga Gadyatskaya\textsuperscript{2} and Fabio Massacci\textsuperscript{2}

\textsuperscript{1} DTU Informatics, Technical University of Denmark, Denmark
\textsuperscript{2} DISI, University of Trento, Italy

Abstract. Java card technology has progressed at the point of running web servers and web clients on a smart card. Yet concrete deployment of multi-applications smart cards have remained extremely rare because the business model of the asynchronous download and update of applications by different parties requires the control of interactions among possible applications after the card has been fielded. The current security models and techniques do not support this type of evolution. We propose in this paper to apply the notion of security-by-contract ($S\times C$), that is a specification of the security behavior of an application that must be compliant with the security policy of the hosting platform. This compliance can be checked at load time and in this way avoid the need for costly run-time monitoring. We show how $S\times C$ can be used to prevent illegal information exchange among applications on a single smart card platform, and to deal with dynamic changes in both contracts and platform policy.
Abstract MadLibs!!

This paper presents a __________ method for __________ (synonym for new) (sciencey verb) the ________________. Using ________________, the (noun few people have heard of) (something you didn’t invent) __________ was measured to be _____ +/- _____ (property) (number) (number) _______. Results show __________ agreement with (units) (sexy adjective) theoretical predictions and significant improvement over previous efforts by __________, et al. The work presented (Loser) here has profound implications for future studies of __________ and may one day help solve the problem of (buzzword) (buzzword) (buzzword)

Keywords: __________, __________, __________ (buzzword) (buzzword) (buzzword) (supreme sociological concern)
The Introduction (1 page)

☑ Briefly introduce the **domain of the problem**

☑ Describe the **problem** (use examples!)

☑ Clearly and explicitly state **your contributions**

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

• Write the **list of contributions**

• **This list 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**”
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.
“Rest of this Paper is…”???

• If possible, 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.

• Someone does not like it, but I do like to have a short Outline of the Paper in the Introduction, after the Contributions:

Outline of the Paper. The rest of this paper is structured [organized] as follows. Section 2 introduces the problem. Section 3 describes ... Section 4 give the details of ... Section 5 presents the related work. Finally, Section 8 concludes with a summary of the main contributions of the paper”.
Structure

- Abstract (~ 4 sentences)
- Introduction and contribution (~ 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)
Wait... Why Not Related Work Yet?!

“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].”

☑️ **Problem 1**: describing alternative approaches gets between the reader and your idea

☑️ **Problem 2**: the reader knows nothing about the problem yet; so your (carefully trimmed) description of various technical tradeoffs is absolutely incomprehensible

I feel tired

I feel stupid
Instead...

☑ Concentrate single-mindedly on a narrative that
  - Describes the problem

  *why is it interesting?*

  - Describes your idea

  - Defends your idea, showing how it solves the problem, and filling out the details

☑ On the way, cite relevant work in passing, but defer discussion to the end
Common (Big) Error: No Idea, Only Details

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

\[
D = \sum_{t=0}^{T} tC F_t D F_t = \sum_{t=0}^{T} tP V_t
\]

Introduce the problem, and your idea, using **EXAMPLES** and only then present the general case!
Conveying 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, he can follow the details (but not vice versa)

☑ Even if he skips the details, he still takes away something valuable
But Don’t Forget Evidence!

☑️ We are talking about **scientific** papers...

☑️ Your introduction makes claims (**list of contributions**)

☑️ The body of the paper MUST provide **evidence** to support each claim

☑️ Evidence can be: analysis and comparison, theorems, measurements, case studies, ...

\[
f(z) = \sum_{m=0}^{\infty} \frac{f^{(m)}(0)}{m!} z^m
\]

\[
= f(0) + f'(0)z + f''(0)\frac{z^2}{2!} + f'''(0)\frac{z^3}{3!} + \cdots
\]

\[
\exp(i\alpha) = 1 + i\alpha - \frac{\alpha^2}{2!} - i\frac{\alpha^3}{3!} + \frac{\alpha^4}{4!} + i\frac{\alpha^5}{5!} - \cdots
\]

\[
= \left(1 - \frac{\alpha^2}{2!} + \frac{\alpha^4}{4!} - \cdots\right) + i\left(\alpha - \frac{\alpha^3}{3!} + \frac{\alpha^5}{5!} - \cdots\right)
\]

\[
= \cos \alpha + i \sin \alpha
\]

Course 02234, DTU, Autumn 2011
Structure

- Abstract (~ 4 sentences)
- Introduction and contribution (~ 1 page)
- The problem (~ 1 page)
- My idea (~ 2 pages)
- The details (~ 5 pages)
- Related work (~ 1-2 pages)
- Conclusion and future work (~ 0.5 pages)
Related Work

- **3rd Fallacy**: to make my work look good, I have to make other people’s work look bad.

  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.
Be Honest!

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!)
Conclusion and Future Work

- Problem
- Summary of contributions
- Strengths
- Weaknesses
- Future work (possible extensions, directions to solve the weaknesses, optimizations, ...)

Course 02234, DTU, Autumn 2011
Other Hints

Helpful Tips
Start Early, Very Early…

☑️ Hastily-written papers get *usually* rejected

☑️ Papers are like wine: they need time to mature

☑️ Collaborate

☑️ Use CVS (or similar tools) to support collaboration
Listening to Your Reviewers

Every review is 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 important!
Listen to Your Reviewers... in Practice

- 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.
Basic (But Still Important) 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
Visual Structure

- sections and sub-sections
- bullets
- italics
- laid-out code

Find out how to draw pictures, and use them!

3.1. ConSpec Syntax

A specification in ConSpec is a non-empty list of rules. Each rule is defined for the specific area of contract, e.g. for the SMS messages, for Bluetooth connections etc. and describes security properties for the given area. Fig. 1 shows a fragment of the ConSpec syntax for specifying one single rule.

Example 3 Fig. 2-3 show the ConSpec specifications of the contract and policy of Ex. 1, respectively.

Example 4 Fig. 4-5 show the ConSpec specifications of the contract and the policy of Ex. 2, respectively.
Example

Figure 3. A heap object

The three cases above do not exhaust the possible forms of \( f \). It might also be a \textit{THUNK}, but we have already dealt with that case (rule \textit{THUNK}). It might be a \textit{CON}, in which case there cannot be any pending arguments on the stack, and rules \textit{UPDATE} or \textit{RET} apply.

4.3 The eval/apply model

The last block of Figure 2 shows how the eval/apply model deals with function application. The first three rules all deal with the case of a \textit{FUN} applied to some arguments:

- If there are exactly the right number of arguments, we behave exactly like rule \textit{KNOWNCALL}, by tail-calling the function. Rule \textit{EXACT} is still necessary — and indeed has a direct counterpart in the implementation — because the function might not be statically known.
- If there are too many arguments, rule \textit{CALLK} pushes a \textit{call}
- The remainder of the object is called the \textit{payload}, and may consist of a mixture of pointees and non-pointees. For example, the object \( \textit{CON}(C, a_1 \ldots a_n) \) would be represented by an object whose info pointer represented the constructor \( C \) and whose payload is the arguments \( a_1 \ldots a_n \).

The info table contains:

- Executable code for the object. For example, a \textit{FUN} object has code for the function body.
- An object-type field, which distinguishes the various kinds of objects (\textit{FUN}, \textit{PAP}, \textit{CON} etc) from each other.
- Layout information for garbage collection purposes, which describes the size and layout of the payload. By “layout” we mean which fields contain pointers and which contain non-pointers, information that is essential for accurate garbage collection.
- Type-specific information, which varies depending on the object type. For example, a \textit{FUN} object contains its arity; a \textit{CON} object contains its constructor tag, a small integer that distinguishes the different constructors of a data type; and so on.

In the case of a PAP, the size of the object is not fixed by its info table; instead, its size is stored in the object itself. The layout of its fields (e.g. which are pointees) is described by the (initial segment of) an argument-descriptor field in the info table of the FUN object which is always the first field of a PAP. The other kinds of heap object all have a size that is statically fixed by their info table.

A very common operation is to jump to the entry code for the object, so GHC uses a slightly-optimised version of the representation in Figure 3. GHC places the info table at the addresses immediately

Course 02234, DTU, Autumn 2011
Use the Active Voice

The passive voice is “respectable” but it DEADENS your paper. Avoid it if possible.

**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
Use Simple, Direct Language

<table>
<thead>
<tr>
<th>NO</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The object under study was displaced</td>
<td>The ball moved sideways</td>
</tr>
<tr>
<td>horizontally</td>
<td></td>
</tr>
<tr>
<td>On an annual basis</td>
<td>Yearly</td>
</tr>
<tr>
<td>Endeavour to ascertain</td>
<td>Find out</td>
</tr>
<tr>
<td>It could be considered that the speed</td>
<td>The garbage collector was really slow</td>
</tr>
<tr>
<td>of storage reclamation left something</td>
<td></td>
</tr>
<tr>
<td>to be desired</td>
<td></td>
</tr>
</tbody>
</table>
If you remember nothing else:

- Identify your **key idea**

- Make your **contributions explicit**

- Use **examples**