How to write a great thesis

Tim Conrad

Based on slides by Simon Peyton Jones, Tine Wirenfeldt Jensen, Tamara O’Connor, Mark Matthews
Demystifying Dissertation Writing

Peg Boyle Single
Why bother?

**Fallacy**

we write theses, papers and give talks mainly to impress others, gain recognition, and get promoted
Good papers and talks are a fundamental part of research excellence (yes, that is also true for your theses!)

Fallacy

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

Why bother?
The purpose of your thesis is...

To convey your idea

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

Everything serves this single goal
The purpose of your thesis 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
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)
• Here’s how my idea compares to other people’s approaches
Brown’s 8 Questions

1. Who are the intended readers?
2. What did you do?
3. Why did you do it?
4. What happened?
5. What do the results mean in theory?
6. What do the results mean in practice?
7. What is the key benefit for readers?
8. What remains unresolved?

(Brown 1994/95)
Structure
Structure

• Abstract
• Introduction
• The problem
• My idea
• The details
• Related work
• Conclusions and further work
Structure (more detailed)

- Title Page
- Table of Contents
- List of Tables
- List of Figures
- Abstract
- Declaration
- Acknowledgments

- Introduction
- Methodology
- Results
- Discussion
- Literature Review
- Conclusions
- Future Work
- Appendices
Structure (how to get there)

- Outline
- Generic thesis structure
- Allocate word count for each section
- Design sub-sections
- Table of contents
- Write in layers
Thesis Map

Success of SE depends on space.

Conceived Methodologies

- The space must reflect the input from many people with different views
- Current methodologies do not address identification a integration of many views

How space tied to analyst's bias and cannot be properly validated

By viewing space as a conversational activity we can model a number of perspectives

This allows us to avoid capturing a reduced perspective before any integration decision

The resulting repository of perspectives acts as a knowledge base to support the negotiation of conflicts

Identifying perspectives

The explicit modelling of perspectives allows us to capture the space building support and validate the resulting space against the various perspectives

Conclusion
A strong title orients readers to your area of work

Effects of Humidity on the Growth of Avalanches

Effects of Humidity on the Growth of Electron Avalanches in Electrical Gas Discharges
A strong title orients readers to your area of work

Studies on the Electrodeposition of Lead on Copper

Effects of Rhodamine-B on the Electrodeposition of Lead on Copper
Structure

- Abstract
- Introduction
- The problem
- My idea
- The details
- Related work
- Conclusions and further work
The abstract

• I usually write the abstract last
• Often used by referee to structure the evaluation report
• 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
SPSE Technique

- **Situation**: Describe the general background / setting in which your research takes place.
- **Problem**: Describe a problem that the research addresses.
- **Solution**: What did you do or try to address this problem?
- **Evaluation**: How did you evaluate the proposed solution and what were the results?
Two Main Approaches

This thesis describes a new inertial navigation system for mapping oil and gas wells. In this thesis, we will compare the mapping accuracy and speed for this new system against the accuracy and speed for conventional systems.

This thesis describes a new inertial navigation system that will increase the mapping accuracy of oil wells by a factor of ten. The new system uses three-axis navigation that protects sensors from high-spin rates. The system also processes its information by Kalman filtering (a statistical sampling technique) in an on-site computer. Test results show the three-dimensional location accuracy is within 0.1 meters for every 100 meters of well depth, an accuracy ten times greater than conventional systems.
Structure

- Abstract
- Introduction
- The problem
- My idea
- The details
- Related work
- Conclusions and further work
The introduction

1. Describe the problem
2. Set the necessary background
3. State your contributions

...and that is all
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 \textit{zipWith}, which zips together two lists, using a function \textit{k} to combine corresponding list elements:

\[
\text{zipWith} :: (a\rightarrow b\rightarrow c) \rightarrow [a] \rightarrow [b] \rightarrow [c]
\]
\[
\text{zipWith} \ k \ [] \ [] = []
\]
\[
\text{zipWith} \ k \ (x:xs) \ (y:ys) = k \ x \ y \ : \ \text{zipWith} \ xs \ ys
\]

Here \textit{k} is an unknown function, passed as an argument; global flow analysis aside, the compiler does not know what function \textit{k} is bound to. How should the compiler deal with the call \textit{k x y} in the body of \textit{zipWith}? It can’t blithely apply \textit{k} to two arguments, because \textit{k} might in reality take just one argument and compute for a while before returning a function that consumes the next argument; or \textit{k} might take three arguments, so that the result of the \textit{zipWith} is a list of functions.
This thesis presents a design for a platinum catalytic igniter in hydrogen-air mixtures. This igniter has application in nuclear reactors. One danger at a nuclear reactor is a loss-of-coolant accident. Such an accident can produce large quantities of hydrogen gas when hot water and steam react with zirconium fuel rods. In a serious accident, the evolution of hydrogen may be so rapid that it produces an explosive hydrogen-air mixture in the reactor containment building. This mixture could breach the containment walls and allow radiation to escape.

Our method to eliminate this danger is to intentionally ignite the hydrogen-air mixture at concentrations below those for which any serious damage might result.
State your contributions

• Write the list of contributions first
• The list of contributions drives the entire thesis: the thesis 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 \texttt{eval/apply}, while those for lazy languages (using call-by-need) have often used \texttt{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, \texttt{eval/apply} seems decisively superior, principally because \texttt{push/enter} requires a stack like no other: stack-walking
State your contributions. CLEARLY!

<table>
<thead>
<tr>
<th>We describe the WizWoz system. It is really cool.</th>
<th>We give the syntax and semantics of a language that supports concurrent processes (Section 3). Its innovative features are...</th>
</tr>
</thead>
<tbody>
<tr>
<td>We study its properties</td>
<td>We prove that the type system is sound, and that type checking is decidable (Section 4)</td>
</tr>
<tr>
<td>We have used WizWoz in practice</td>
<td>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.</td>
</tr>
</tbody>
</table>
No “rest of this thesis is...”

• Not:

  “The rest of this thesis 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 thesis, and therefore forward reference every important part.
Possible sections in this chapter

1. Background
2. Research Problem
3. Methodology (or method to be used)
4. Thesis Roadmap / Overview / Outline
5. Definitions
6. Orthogonal Issues
7. Conclusions
Structure

- Abstract
- Introduction
- The problem
- My idea
- The details
- Related work
- Conclusions and further work
No 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].
No related work yet

• **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
Instead...

Concentrate single-mindedly on a narrative that

- Describes the problem, and why it is 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
The payload of your thesis

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 epi-modulus $j$, such that $p_j < p_i$.

- Sounds impressive...but
- Sends readers to sleep
- In a thesis you MUST provide the details, but FIRST convey the idea
The payload of your thesis

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`:

```plaintext
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`.

The definition of `gsize` refers to the operation `gmapQ`, which is a method of the `Data` class:

```plaintext
class Typeable a => Data a where
   ...other methods of class Data...
gmapQ :: (forall b. Data b => b -> r) -> a -> [r]
```
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, she can follow the details (but not vice versa)
• Even if she skips the details, she still takes away something valuable
Solving the Research Question

• You need to convince the reader that the research question has been resolved
• Only discuss your research that pertains to the resolution of the problem
• If you worked on techniques that were dead-ends, don't bother including these in the discussion (unless it is somehow relevant to the solution).
Evidence

• Your introduction makes claims
• The body of the thesis 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
Solving the Research Question

• Record any limitations, assumptions, etc. in methods used

• Report exactly under what conditions the results were obtained

• If you wrote computer programs to solve the problem
  -> record hardware, software versions, language versions, operation system versions, any relevant environmental configurations
Strategies for the main part

Choose a logical strategy

Make sections and subsections

Heading
Subheading
Subheading

Heading
Subheading
Subheading
Subheading

Heading
Strategies for the main part

Chronological

Spatial

[Maizels, 2001]

[Pratt & Whitney, 2000]
Strategies for the main part

Parallel Parts

Flow
Non-Parallel  Parallel
Non-Descriptive Descriptive

Introduction
Past Designs for Particle Beam Fusion

Background
New Design for Particle Beam Fusion

Marx Generators Charging Marx Generators
Line Pulse Forming Line Pulse
Beam Generation Generating Particle Beam
Transporting Beam Transporting Particle Beam
Pellets Irradiating Deuterium-Tritium Pellets

Results
Results of New Design

Conclusions
Conclusions and Recommendations
New Design for Particle Beam Fusion

Charging Marx Generators

Generating Particle Beam

Pellets
New Design for Particle Beam Fusion

Charging Marx Generators

Generating Particle Beam

Irradiating Deuterium-Tritium Pellets
Secondary Structure/Headings

Performance of the Solar One Receiver
Introduction
Steady State Efficiency
Average Efficiency
Start-Up Time
Operation Time
Operation During Cloud Transients
Panel Mechanical Supports
Tube Leaks
Conclusion

Performance of the Solar One Receiver
Introduction
Receiver’s Efficiency
Steady State Efficiency
Average Efficiency
Receiver’s Operation Cycle
Start-Up Time
Operation Time
Operation During Cloud Transients
Receiver’s Mechanical Wear
Panel Mechanical Supports
Tube Leaks
Conclusion
Structure

- Abstract
- Introduction
- The problem
- My idea
- The details
- Related work
- Conclusions and further work
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 thesis

- 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 thesis

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
• Introduction
• The problem
• My idea
• The details
• Related work
• Conclusions and further work
Strong ending

Conclusions
Analysis of Results

Analyze results from overall perspective

Future Perspective

Several options:
Make recommendations
Discuss future work
Repeat limitations
Language & Style
Visual structure

• Give strong visual structure to your text using
  • sections and sub-sections
  • bullets
  • italics
  • laid-out code

• Find out how to draw pictures, and use them
Visual structure

![Diagram of a heap object]

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

- If there are exactly the right number of arguments, we behave exactly like rule \texttt{EXACTCALL}, by i.e. calling the function. Rule \texttt{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 \texttt{CALLX} pushes a call remainder of the object is called the \textit{payload}, and may consist of a mixture of pointers and non-pointers. For example, the object \( \texttt{CON}(C, a_1 \ldots a_n) \) 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 \texttt{FUN} object has code for the function body.
- An object-type field, which distinguishes the various kinds of objects (\texttt{FUN}, \texttt{PAP}, \texttt{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 \texttt{FUN} object contains its arity, a \texttt{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 \texttt{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 pointers) is described by the (initial segment of an argument-descriptor field in the info table of the \texttt{FUN} object which is always the first field of a \texttt{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 \texttt{GHC} uses a slightly-optimised version of the representation in Figure 3. \texttt{GHC} places the info table at the addresses immediately
# Use the active voice

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

<table>
<thead>
<tr>
<th>NO</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>It can be seen that...&lt;br&gt;34 tests were run&lt;br&gt;These properties were thought desirable&lt;br&gt;It might be thought that this would be a type error</td>
<td>We can see that...&lt;br&gt;We ran 34 tests&lt;br&gt;We wanted to retain these properties&lt;br&gt;You might think this would be a type error</td>
</tr>
</tbody>
</table>

“We” = you and the reader

“We” = the authors

“You” = the reader
Use simple, direct language

<table>
<thead>
<tr>
<th>NO</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The object under study was displaced horizontally</td>
<td>The ball moved sideways</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 of storage reclamation left something to be desired</td>
<td>The garbage collector was really slow</td>
</tr>
</tbody>
</table>
Rule of Three

• Within each chapter, repeat 3 times
  • Intro. *We will show* ..
  • Body. *Show them* ..
  • Concl. *We have shown* ..

• Within thesis, repeat contributions 3 times
  • Intro chapter
  • Main chapters
  • Conclusion chapter
Link chapters together

Chapter N
Introduction N.1
As has been discussed in the previous chapter ...

Conclusion N.X
In this chapter, the following topics were discussed...equipped with this information, in the next chapter it will be investigated how to apply this in ...
More...
The process

• Start early. Very early.
  • A hastily-written thesis decreases your overall mark.
  • Theses are like wine: they need time to mature
• Use a backup system (SVN / GitHub)
Getting help

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

• Experts are good
• Non-experts are also very good
• Each reader can only read your thesis for the first time once! So use them carefully
• Explain carefully what you want ("I got lost here" is much more important than "wibble is mis-spelt").
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, 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 thesis so that X is apparent even to the stupidest reader.
• Thank them warmly. They have given up their time for you.
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, a written-out proof or a script)
  in an appendix
• Always use a spell checker
What are examiners looking for?

- Review of literature
- Methodology
- Presentation of Results
- Application to biology
- Discussion and Conclusions
Tips for successful writing

1. Plan to write regularly
2. Make a time plan and stick to it
3. Write up section as soon as it’s ready
4. Stop writing at a point where you could go on – makes it easier to start next time!
5. Decide where and when best for you
6. Don’t write when exhausted
7. Seek support
Summary

If you remember nothing else:
• Identify your key idea
• Make your contributions explicit
• Use examples

http://www.phrasebank.manchester.ac.uk/