Working on your IRP: Argument, identifying claims

Simon Peyton Jones
Microsoft Research Cambridge
with Alan Bundy, University of Edinburgh

Revised and Extended for IRP: February 2011
Jon Oberlander, University of Edinburgh

Structure of this tutorial

1. Writing a proposal
   - Problems, Ideal proposals
1. Hypotheses in Informatics
   - Identifying Claims

Reminder: Advice

- Identify hypothesis and how to evaluate it
- Plan research programme
  - Break project into work-packages
  - Gauge duration, deliverables and dependencies
- Motivation: significance, feasibility, novelty
- Pace yourself
  - Leave time for feedback and correction

Reminder: Structure

- Motivation:
  - aims and objectives, hypothesis, timeliness, significance, feasibility, novelty, beneficiaries
- Background material
  - use your IRR if you can
- Methodology and techniques to be used
- Metrics for evaluation
- Outcomes
  - application? experimental results?
- Research plan
  - usually in the form of a Gantt chart

Reminder: Common problems

- Hypothesis is unclear, ill-formed, or blatantly wrong
- Project attempts to solve a non-problem
- Assuming you will succeed where others have failed
- Insufficient detail to assess outcomes
- Unaware of related research
- Bad presentation, incomprehensible report
- KISS = Keep It Simple, Student (words to live by)

The vague proposal

1. I want to work on better type systems for functional programming languages
2. **Give me the money**

You absolutely must identify the problem you are going to tackle

Identifying the problem

- What is the problem?
- Is it an **interesting** problem? That is, is it research at all?
- Is it an **important** problem? That is, would anyone care if you solved it? (jargon: “impact”)
- Having a "customer" helps: someone who wants you to solve the problem

The aspirational proposal

1. I want to solve the problem of avoiding deadlocks and race conditions in concurrent and distributed programs
2. **Give me the money**

- **It is easy to identify an impressive mountain**
- **But that is not enough! You must convince your reader that you stand some chance of climbing the mountain**

Climbing the mountain

Two sorts of evidence

1. You absolutely must say what is the **idea** that you are bringing to the proposal.
2. Explain modestly but firmly why **you** are ideally equipped to carry out this work.
   (NB: not enough without (1))

1. Your idea

- Give real technical "**meat**", so an expert reader could (without reading your doubtless-excellent papers) have some idea of what the idea is
- Many, many grant proposals have impressive sounding words, but lack almost all technical content. Reject!

1. Your idea

Offer **objective evidence** that it's a **promising** idea:

- Results of preliminary work
- Prototypes
- Publications
- Applications

Strike a balance: you don't want the reader to think "they've already solved the problem".

Your message

We are ideally placed to do this timely research because

- We have an idea
- Our preliminary work shows that it’s a promising idea
- We are the best in our field

The I'll-work-on-it proposal

1. Here is a (well-formulated, important) problem
2. Here is a promising idea
3. We're a world-class team
4. We'll work on it
5. **Give us the money**

The key question
How would a reviewer know
if your research had succeeded?
Jargon: "aims, objectives"

Suspicious phrases

- “Gain insight into…”
- “Develop the theory of…”
- “Study…”

The trouble with all of these is that there is no way to distinguish abject failure from stunning success.

Good phrases

- “We will build an analyser that will analyse our 200k line C program in less than an hour”
- “We will build a prototype walkabout information-access system, and try it out with three consultants in hospital Y”

The most convincing success criteria involve those “customers” again

Related work

- **Goal 1**: demonstrate that you totally know the field. Appearing ignorant of relevant related work is certain death.
- **Goal 2**: a spring-board for describing your promising idea
- But that is all! Do not spend too many words on comparative discussion. The experts will know it; the non-experts won't care.

Methodology/plans

- “Methodology”, or describing your step-by-step plans, is usually over-stressed in my view.
- Concentrate on (a) your idea, and (b) your aims/objectives/success criteria. Then the “methodology” part writes itself.

The ideal proposal

1. Here is a well-defined problem
2. It's an important problem (evidence…)
3. We have a promising idea (evidence…)
4. We are a world-class team (evidence…)
5. Here is what we hope to achieve
6. Here is how we plan to build on our idea to achieve it
7. **Give us the money. Please.**

Help each other

Ask others to read your proposal critically
Revise, and ask someone else
Repeat

- Cheap: what someone thinks after a 10-minute read is Really Really Important
- Informative: after reading 20 proposals by others, you'll write better ones yourself. Much better proposals.
- Effective: dramatic increases in quality. There is just no excuse for not doing this.

Hypotheses in Informatics

- Claim about task, system, technique or parameter, *e.g.*:

  "All techniques to solve task X will have property Y."

  "System X is superior to system Y on dimension Z."

  "Technique X has property Y."

  "X is the optimal setting of parameter Y."

- Properties and relations along scientific, engineering or cognitive science dimensions.

- Theoretical or experimental investigation.

Rarely explicitly stated.

Typical claims in Informatics

X is better than Y on task Z along some dimension W

- What kind of things are X and Y?
  - system?
  - technique?
  - parameter?
- What is task Z?
- What is the dimension W?
  - behaviour, coverage, efficiency, usability, dependability, maintainability

For keyword-based searches in medical databases, Pseudo-Relevance Feedback will provide better search results than Topic Modeling as measured by mean average precision of the ranked list.

13

Scientific dimensions

Behaviour: *the effect or result of the technique,*

correctness *vs* quality,

need external 'gold standard';

Coverage: *the range of application of the technique,*

completeness *vs* partial;

Efficiency: *the resources consumed by the technique,*

e.g. run time or space used,

usually as approx. function, *e.g.* linear, quadratic, exponential, terminating.

sometimes mixture of above.

property *vs* comparative relation.

task *vs* systems *vs* techniques *vs* parameters.

Engineering dimensions

Fitness: *how well it meets user requirements.*

Usability: *how easy to use?*

Dependability: *how reliable, secure, safe?*

Maintainability: *how evolvable to meet changes in user requirements?*

Scalability: *whether it still works on large/complex examples?*

Cognitive science dimensions

External: *match to external behaviours*,

both correct and erroneous.

Internal: *match to internal processing.*

clues from *e.g.* protocol analysis.

Adaptability: *range of occurring behaviours modelled*

... and non-occurring behaviours.

Evolvability: *ability to model process of development.*

All this to some level of abstraction.