Reasoned Modelling

Exploiting the Synergy between Reasoning & Modelling

Andrew Ireland[1]

Joint work with:

Gudmund Grov[2] Maria Teresa Llano[1] Alison Pease[2]

[1]School of Mathematical and Computer Sciences
Heriot-Watt University

[2]School of Informatics
University of Edinburgh

Motivations

"... develop a program and its proof hand-in-hand, with the proof ideas leading the way!"

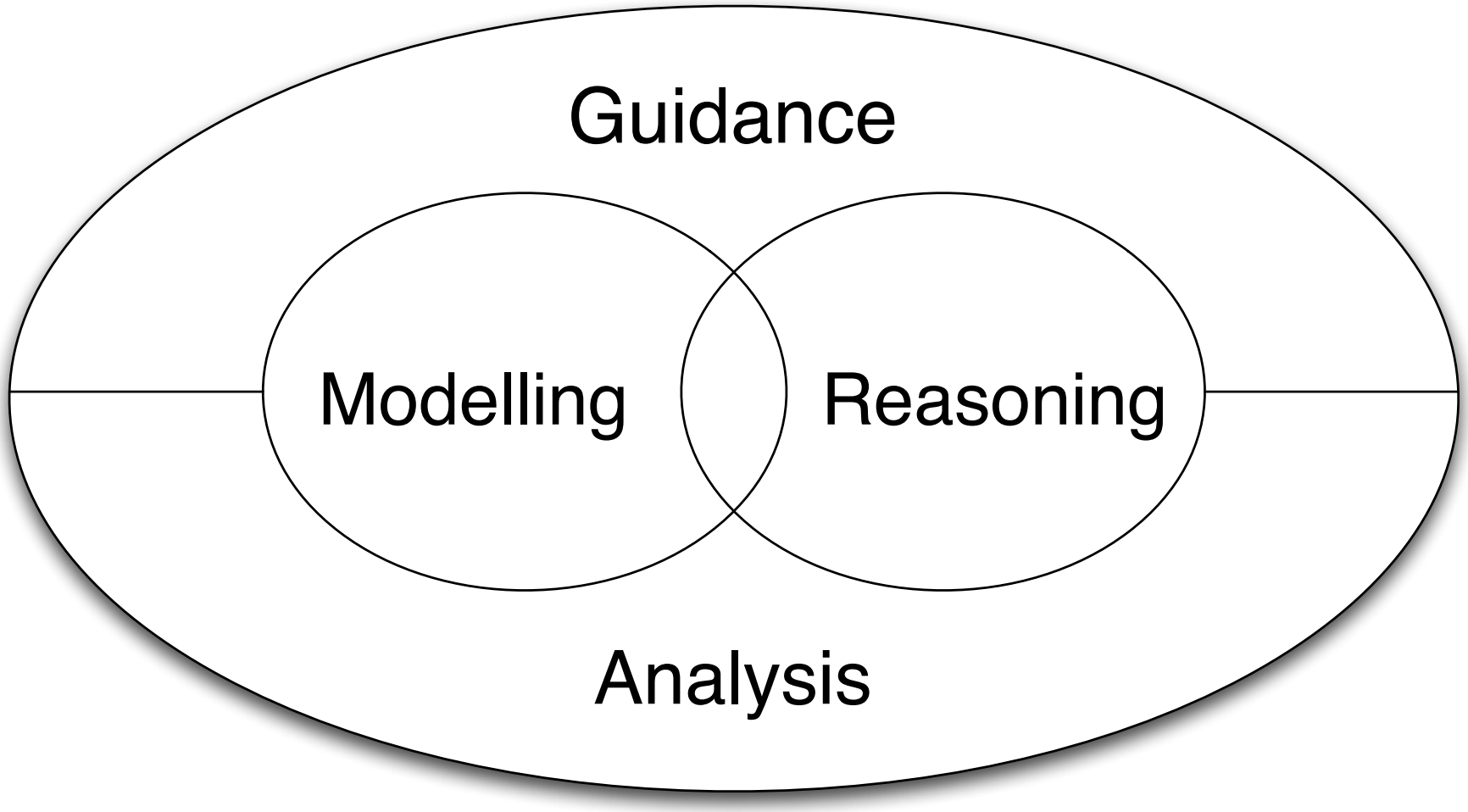
(David Gries, 1981)

- While the rigour of building formal models brings significant benefits, formal reasoning remains a major barrier to the wider acceptance of formalism within the development of software intensive systems.
- We aim to abstract away from the complexities of low-level proof obligations, providing high-level modelling guidance – *we call this* **reasoned modelling**.

"... with the proof ideas being translated into principled design level guidance!"

- Accessibility and productivity – *allow smart designers to make better use of their time.*
- It's NOT about hacking a design in order to complete the proofs.

Reasoned Modelling - Big Picture



Reasoned Modelling - An Event-B Perspective

Reasoned modelling critics: Combining proof-failure analysis and modelling priorities (meta-data) in order to constrain the search for design level guidance, *e.g. a cars braking system should be given priority over its cruise control.* [J. SCP 2011]

Refinement plans: Combining common patterns of refinement with their associated proof patterns in order to constrain the search for design level guidance, *e.g. splitting the atomicity of a data transfer or replacing a set partition by a function.* [ABZ 2012]

HRemo: Combining design level simulations (model execution) with proof-failure analysis and automated theory formation in support of invariant discovery, *i.e. gluing invariants.* [Refine 2011]

The Challenge of Abstraction

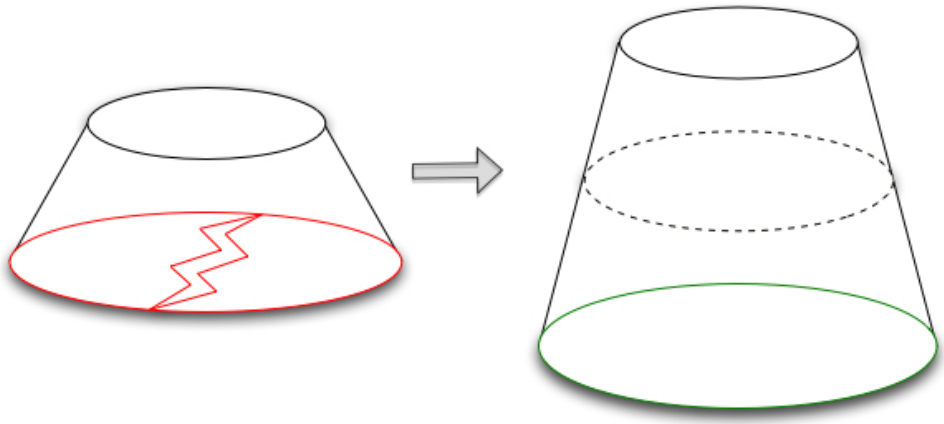
> *"... computer science is a science of abstraction – creating, the right model for thinking about a problem and devising the appropriate mechanizable techniques to solve it."*
>
> (Alred V. Aho & Jeffrey D. Ullman, 1995)

- Given the *right* granularity of abstract models, the complexity of a design can be mastered, both in terms of generating formal *guarantees* as well as *explaining* the key "design ideas".
- But getting it "right" on industrial scale is a challenge!
- Our current work has led us to consider how one could support the creation of abstractions that:
  - Improve the *explanatory* power of the design representation.
  - Reduce the complexity of constructing formal *guarantees* (proofs).
  - Strengthen the links between the requirements and the design.

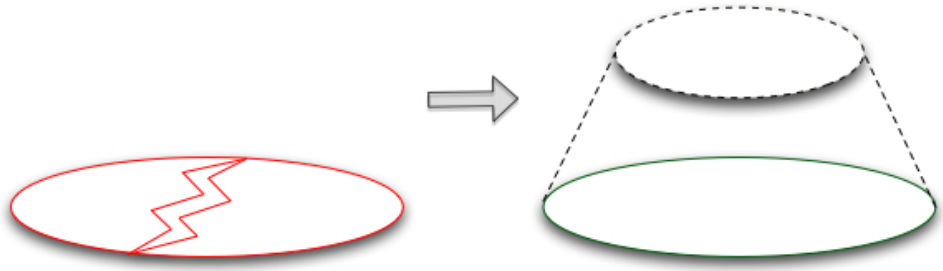
We see our refinement plans as providing a starting point ...

Abstraction and Refinement Based Modelling

- Creating intermediate layers:



- Creating a more abstract starting point:



- Failure preserving abstractions, i.e. making your failures more understandable – *a basic "method" within the ACL2 community.*
- How to get started?

More AI4FM Synergies?

- User scenarios and simulations of early "design ideas" as a stepping stone to a complete design
  - *can machine learning and computational creativity help, e.g. the HR Automated Theory Formation system?*
- Bridging the gap between informal requirements and formal concepts within our design models
  - *can AI techniques help?*