Engineering Transparent Simulations for Science

Paul Andrews

York Centre for Complex System Analysis
and Department of Computer Science
University of York

Background

- CoSMoS project funded by EPSRC to develop tools and techniques for complex systems simulation
- Computer systems engineering
  - Decades of best practice from software development
- Range of biological and social science case studies
- Agent based approaches, but should apply across modelling and simulation techniques

Overview

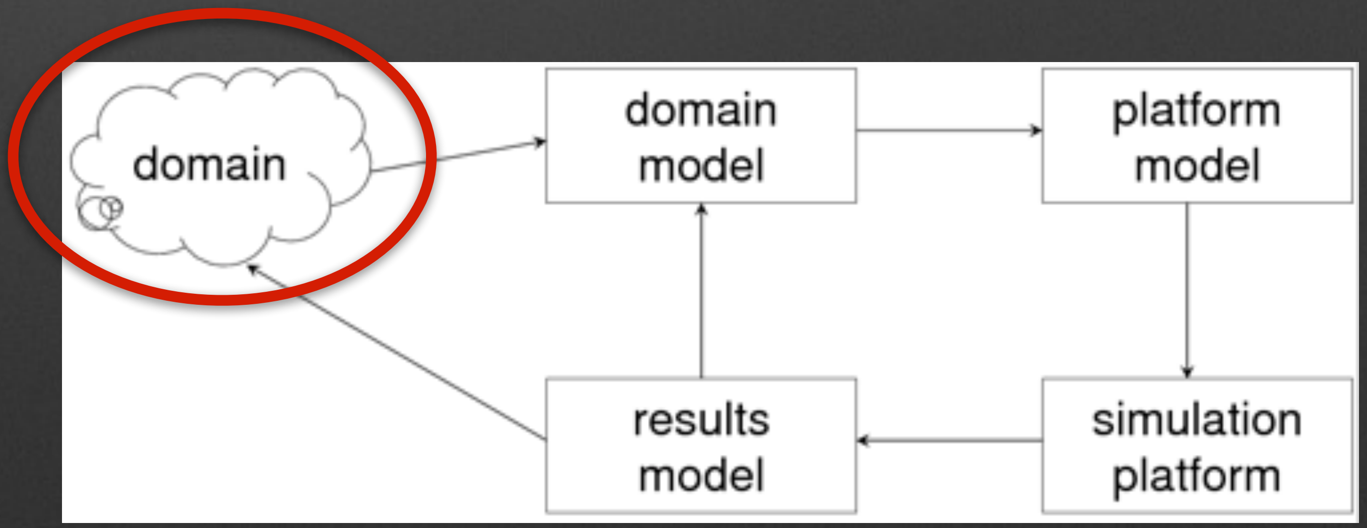
- Summary of the outputs from the CoSMoS project and team
- CoSMoS models: 4 boxes and a blob
- CoSMoS patterns and process
- Validation and simulations as scientific instruments
- Calibration and Argumentation

Models for Simulation

- Models used to capture, communicate and reason about scientific simulation
  - Establish the scientific basis of research
  - Development and use of simulation platform
- Improve scientific rigour
  - Open to review and challenge
  - Provides a basis for scientific reproducibility

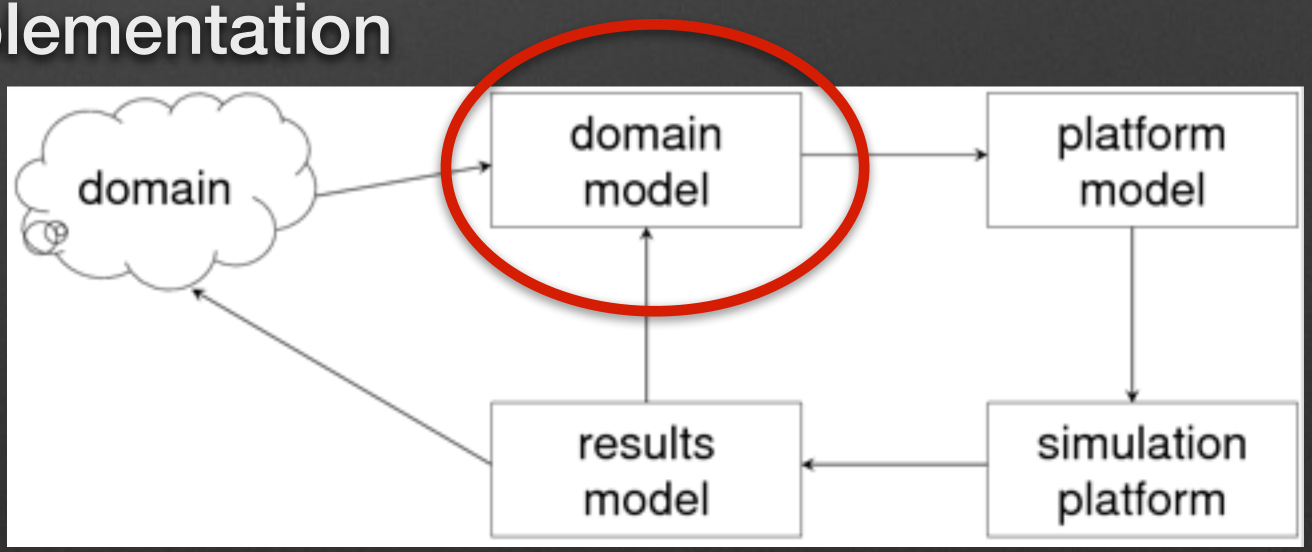
CoSMoS Models

- A language to express your model concepts and simulation results
- The **Domain** represents the real-world system of investigation
  - A domain experts view



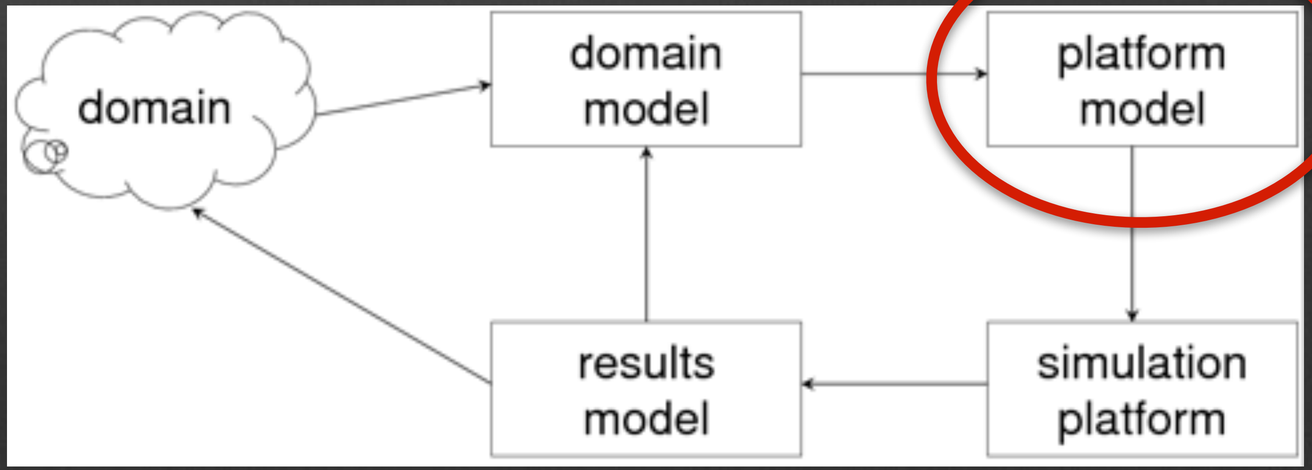
Domain Model

- Explicitly captures domain knowledge
    - Describes structures, behaviours and interactions
    - “Suitable” level of detail and abstraction
- Ideally developed with domain experts
    - Exchange and discuss domain understanding
- Distinguishes science from implementation



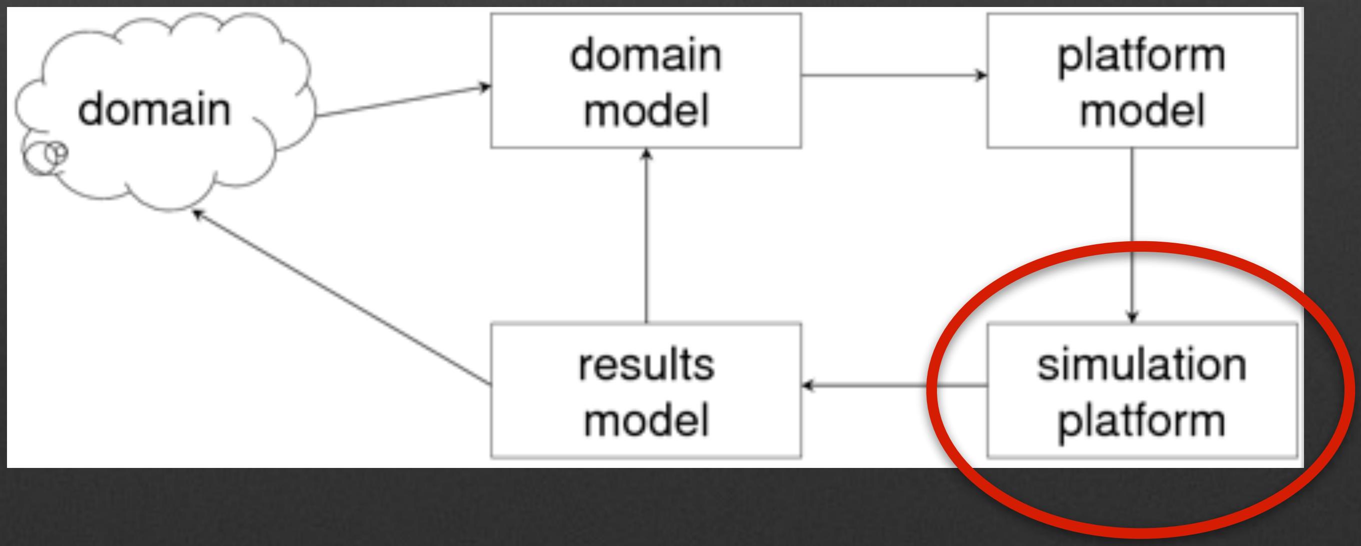
Platform Model

- Engineering derivation from domain model:
  - Details implementation of domain model concepts
- Remove, add, simplify:
  - Remove hypothesised mechanisms (don't code in the answer)
  - Add components to observe, interact, and record (instrumentation)
  - Abstract or simplify for efficient implementation



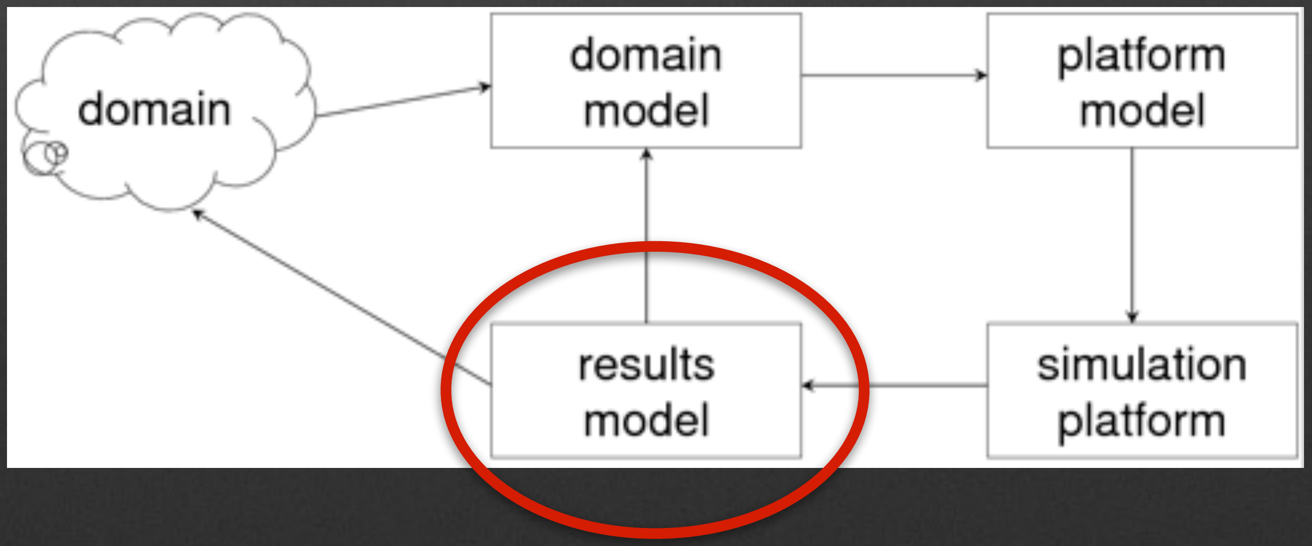
Simulation Platform

- What you run your experiments on
  - Hardware + OS + software libraries + your code
- Software libraries and code implement the platform model
- Understand what your libraries do
  - Pseudo-random number generation

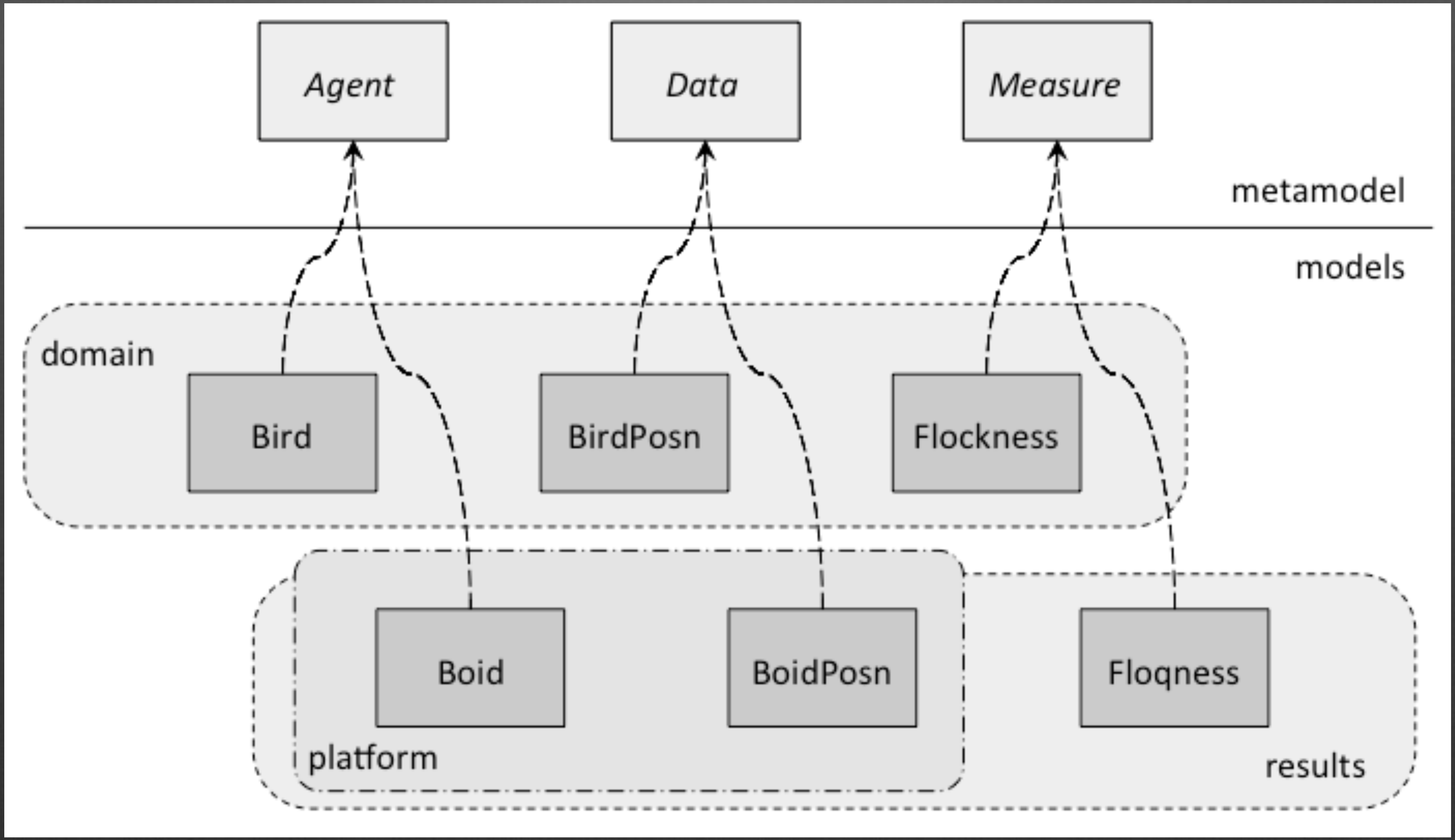


Results Model

- Analysis of simulation experiments
  - Observations, screenshots, dynamic sequences, raw output data, result statistics
  - Basis for interpreting simulation results
- Compare to domain model to establish suitability of simulation platform
- Provide insight for new experimentation, real or simulated



Toy Example



Thanks to Susan Stepney

CoSMoS Patterns

- Software engineering best practice can be captured using *patterns*:
  - Core solutions to reoccurring types of problem
- CoSMoS patterns provide advice on how navigate the CoSMoS models just described
- Shameless plug: Book

CoSMoS Process Pattern

- Iterate through three phases: discovery, development and exploration
- Discovery:
  - Deciding what simulation platform needs to be built
- Development:
  - Building the simulation platform
- Exploration:
  - Using the platform to perform simulation experiments

Discovery Phase

- Identify the research context:
  - Goals, assumptions, simulation purpose, success criteria
- Define the domain
- Identify expected behaviours (hypotheses)
- Build a domain model
- Prototype simulation platform
- Argue appropriate instrument designed

Development Phase

- Develop platform model
  - Changes from domain model, assumptions, abstractions, simplifications
- Develop simulation platform
  - Good software engineering: choosing appropriate language, testing, calibration
- Argue appropriate instrument built

Exploration Phase

- Revisit research context (are we still on track)
- Develop results model
  - Simulation experiments to build a picture of what the simulator's behaviour
- Perform simulation experiments
- Argue instrument used appropriately

Not a Tick List

- Don’t be a slave to the process
    - React to what your investigations reveal
- Sometimes discovery phase is all you need
    - Insight gained just by sitting down and trying to produce a logically consistent description (the domain model) of the system.
    - Being explicit highlights misunderstandings and inconsistencies

Validation and Confidence

- Validation is not absolute
  - Degrees of confidence in applying simulation insights to real-world
- Confidence in the modelling process
  - Argumentation techniques
- Confidence in the implementation
  - Good software engineering practices
- Confidence in the simulation outputs
- Statistical techniques

Transparent Simulations

- Communicating confidence to the wider community
- Increase impact of simulation based science?
- Open code and beyond
  - An analogy to scientific instruments

Open Code

- Computational science has added new layer of inaccessibility
- Reproducibility is a key axiom of science
  - Best achieved through understanding the code
- But:
  - Code is just the end product of a process of science and engineering
  - Understanding issues of construction and use can help interpretation and presentation of simulation results

Ince et al., The case for open computer programs, *Nature*, 462(26), 2012

Scientific Instruments

- Enhance the range of our natural abilities
- Specialised instruments require careful use
  - Idiosyncrasies can produce spurious data
- Understanding the instrument's operation (both theory and implementation) aids usage
  - Identify operational artefacts
  - Know the boundaries of operation

Humphreys, *Extending Ourselves,* Oxford University Press, 2004

Simulations as Instruments

- All instruments encode some model of understanding of a target domain
  - Computational techniques allows us to encode part or all of this model using logic
- Computer simulators can be viewed as bespoke scientific instruments
  - Expose them to the same standards of engineering

Calibration

- Establishes relationship between instrument outputs and system under observation
- Affords confidence in using the instrument
  - Identifies scope (what it can be used for), accuracy, precision and resolution
- Calibrating simulations can be difficult
  - Finding suitable data
  - Each simulator will require different set of calibration requirements
  - Changes to simulator may require complete recalibration

Calibration through Statistics

- Establish baseline simulation behaviour with domain expert
- Discover representative results
  - What is explained by stochastic variation
  - What is due to a property of the underlying model
- Interpret simulation results in terms of the domain metrics
- SPARTAN

Argumentation

- Bringing everything together
- High integrity systems engineering
- Capture confidence arguments
  - Suitable model, implementation, results
- Expose research to review
- Document assumptions, simplifications, abstractions
- Artoo and the Goal-Structuring Notation

Some Advice

- All models are wrong, but what is your model?
  - Your model of the science and your code are not the same
- You don’t get the “physics for free”
- Try to have both science and engineering expertise
  - If it’s just you, separate the activities
- Different types of assumption made throughout process of building models, simulators and running simulations
- Openness and communication are vital

Acknowledgements

- Many, including:
  - Kieran Alden, Jim Bown, Teodor Ghetiu, Tim Hoverd, Fiona Polack, Mark Read, Carl Ritson, Adam Sampson, Susan Stepney, Jon Timmis, Peter Welch, Alan Winfield

- www.york.ac.uk/computational-immunology/software/