sicsa*
The Scottish Informatics & Computer Science Alliance

THE UNIVERSITY OF EDINBURGH

Scientific Methods in Research

SICSA PHD CONFERENCE 2017

UNIVERSITY OF DUNDEE

ALAN BUNDY

School of informatics

UNIVERSITY OF EDINBURGH

What is Computing?

Informatics: The study of the structure, behaviour, and interactions of both natural and artificial computational systems

What are the Big Computing Questions?

- What is the nature of computation/information?
- What is mind?
- How can we build useful ICT products?

Exploration of Technique Space

Computing as the space of computational techniques.

Job of Computing to explore this space.

- Which techniques are good for which tasks?
- What are properties of these techniques?
- What are relationships between these techniques?

What are Computing Techniques?

Information Representation: e.g. databases, hash tables, production rules, neural nets.

Algorithms: e.g. quick sort, depth-first search, parser.

Architectures: e.g. von Neumann, parallel, agents.

Software Engineering Processes: e.g. extreme programming, knowledge acquisition/requirements capture.

Theories: e.g. denotational semantics, process algebras, computational logics, hidden Markov models.

Exercise: Computing Techniques

What additional Computing techniques can you think of?

- Information Representation?
- Algorithms?
- Architectures?
- Software Engineering Processes?
- Theories?
- Other kind?

The Space of Computing Techniques

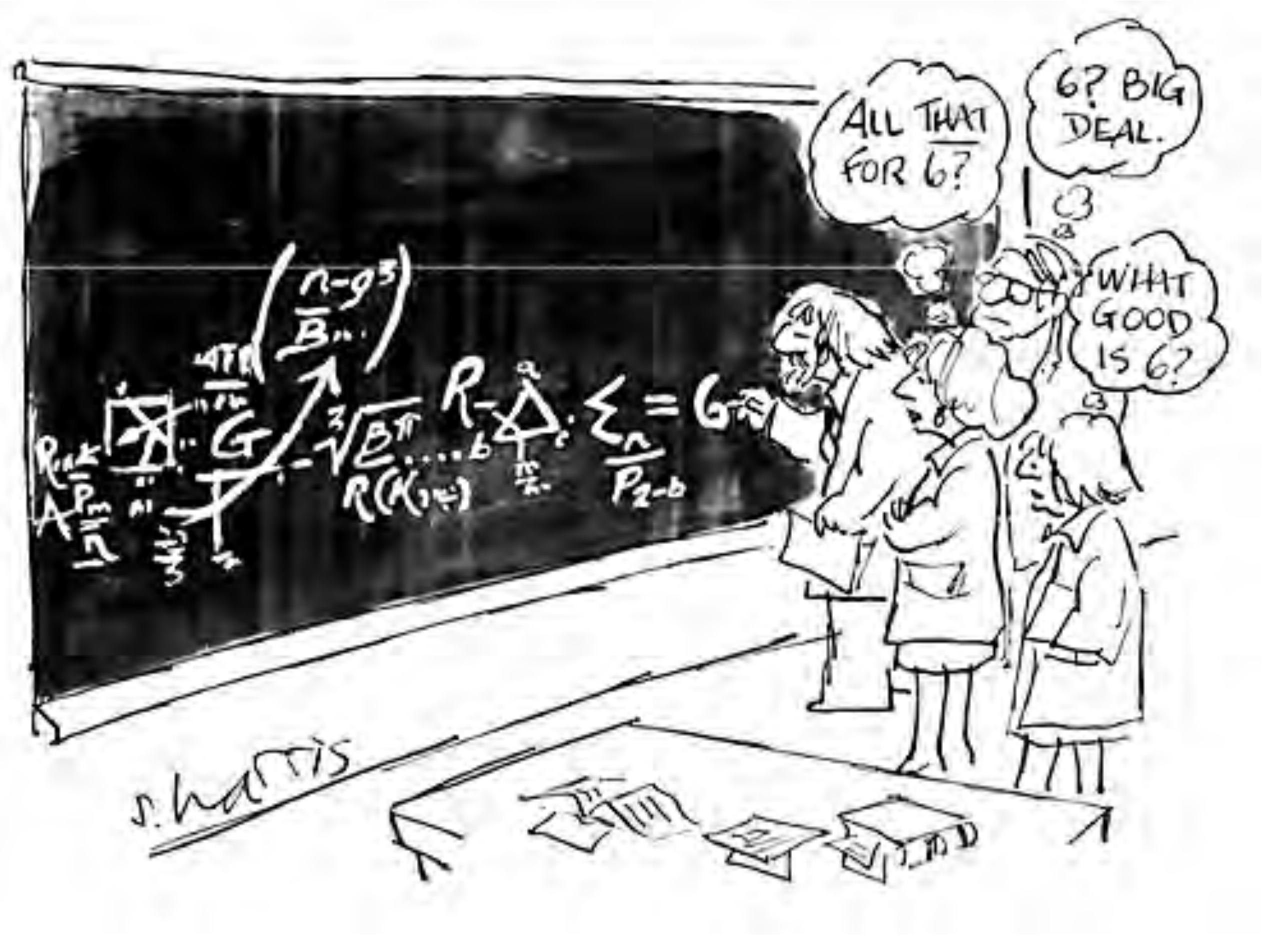
Multi-dimensional space of techniques,
- linked by relationships.

Rival techniques for same task,
- with tradeoffs of properties.

Complementary techniques which interact.

Build systems from/with collections of techniques

The Significance of Research



Importance of Hypotheses

Science and engineering proceed by

- the formulation of hypotheses
- and the provision of supporting (or refuting) evidence for them.

Computing should be no exception.

But the provision of explicit hypotheses in Computing is rare.

This causes lots of problems.

My mission – to persuade you to rectify this situation.

Problems of Omitting Hypotheses

Usually many possible hypotheses.

Ambiguity is major cause of referee/reader misunderstanding.

Vagueness is major cause of poor methodology:

- Inconclusive evidence;
- Unfocussed research direction.

Exploration of Techniques Space

Invention of new technique,

Investigation of technique,

- e.g. discovery of properties of, or relationships between, techniques.

Extension or improvement of old technique,

New application of a technique,

- to artificial or natural systems.

Combine several techniques into a system.

Hypotheses in Computing

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.

Ideally, with the addition of a ‘because’ clause.

Properties and relations along scientific, engineering computational modelling dimensions.

May be several hypothesis in each publication.

Rarely explicitly stated

Scientific Dimensions 1

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

- correctness *vs* quality,
- need external 'gold standard';

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

- complete *vs* partial;

Efficiency: *the resources consumed by the technique,*

- e.g. time or space used,
- usually as approx. function, e.g. linear, quadratic, exponential, terminating.

Scientific Dimensions 2

Sometimes mixture of dimensions,

- e.g., behaviour/efficiency poor in extremes of range.

Sometimes trade-off between dimensions,

- e.g., behaviour quality vs time taken.

Property *vs* comparative relation.

Task *vs* systems *vs* techniques *vs* parameters.

Engineering Dimensions

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 complex examples?*

Cost*: In £s or time of development, running, maintenance, etc.*

Portability*: interoperability, compatibility.*

Computational Modelling 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 not modelled.*

Evolvability: *ability to model process of development.*

All this to some level of abstraction.

Exercise: Hypotheses

What Computing hypotheses can you think of?

Choose system/technique/parameter setting.

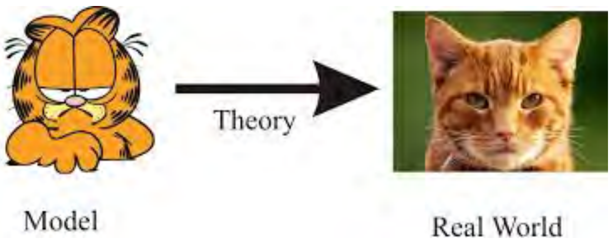
Choose science/engineering/computational modelling dimensions.

Choose property or relation.

Has property or is better than rival on property?

Other?

Theoretical Research



Use of **mathematics** for definition and proof.
- or sometimes just reasoned argument.

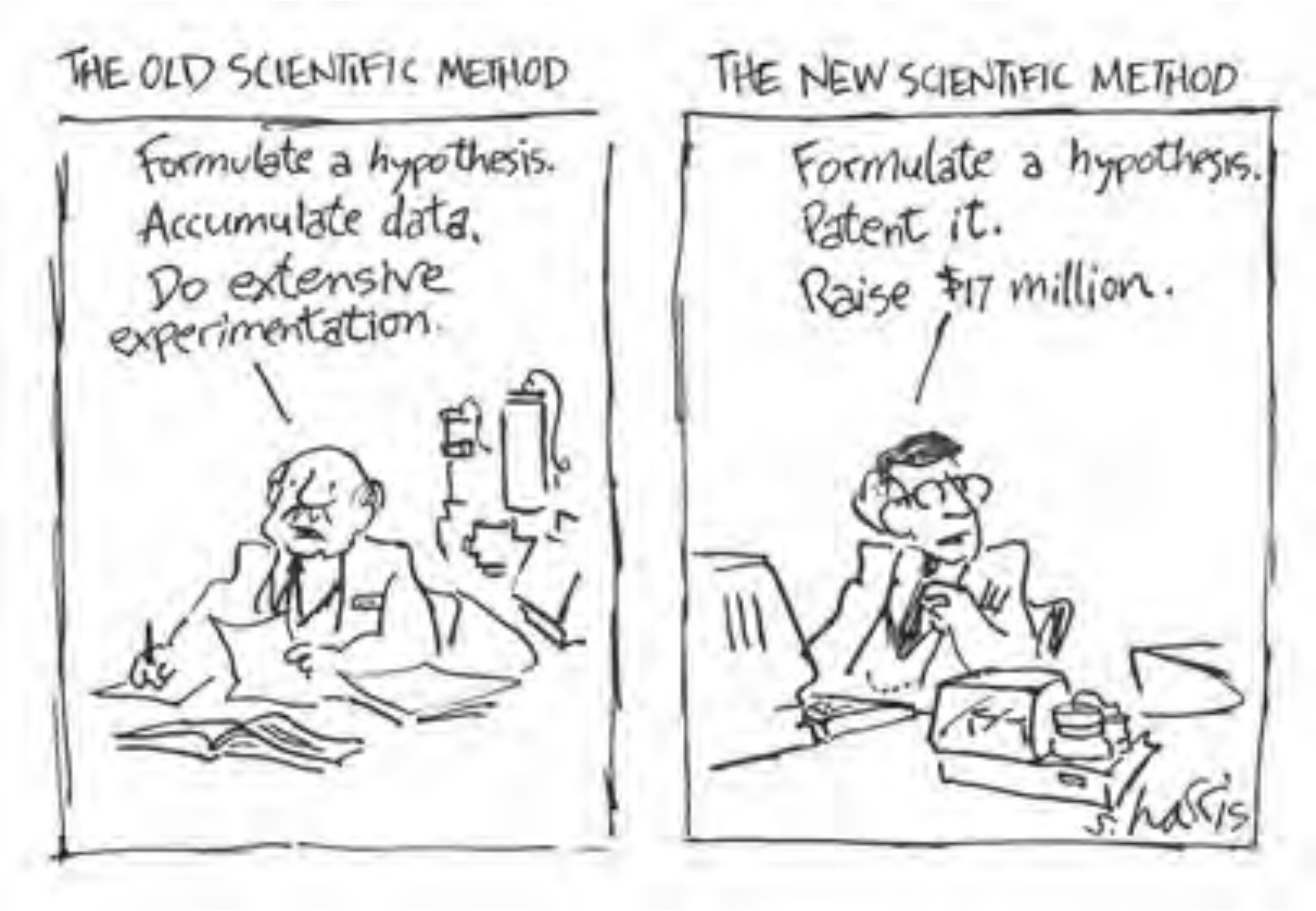
Applies to task or technique.

Theorem as hypothesis; **proof** as evidence.

Advantages:
- Abstract analysis of task;
- Suggest new techniques, e.g. generate and test;
- Enables proof of general properties/relationships,
  - cover potential infinity of examples;
  - Suggest extensions and generalisations;

Disadvantage:
- Sometimes difficult to reflect realities of task.

Experimentation



Experimental Research

Kinds:

- exploratory vs hypothesis testing.

Generality of Testing:

- test examples are representative.

Results Support Hypothesis:

- and not due to another cause.

How to Show Examples Representative

Distinguish development from test examples.

Use lots of dissimilar examples.

Collect examples from an independent source.

Use the shared examples of the field.

Use challenging examples.

Use acute examples

How to Show that Results Support Hypothesis

Vary one thing at a time,

- then only one cause possible.
- Unfortunately, not always feasible.

Analyse/compare program trace(s),

- to reveal cause of results.

Use program analysis tools,

- e.g. to identify cause/effect correspondences

Hypotheses must be Evaluable

If hypothesis cannot be evaluated then fails Popper's test of science.

Obvious hypothesis may be too expensive to evaluate,

- e.g. programming in MyLang increases productivity,

Replace with evaluable hypothesis:

- Strong typing reduces bugs.
- MyLang has strong typing.

Different Views of Computing

Computing as science.

Computing as engineering.

Computing as mathematics.

Computing as art.

Computing as Science

What is the nature of computation & information?

- A "science of the artificial" [Herb Simon]
- Others: economics, linguistics, sociology, ...

Experimental studies of algorithms, architectures and systems.

- Pose hypotheses and test them.

Computing as Engineering

How can we build useful ICT products?

- Design, development, deployment and maintenance.
- Hardware and/or software.

Still make and test claims.

- This software engineering methodology facilitates fast development of dependable systems.
- This programming language eliminates certain kinds of bug.

Computing as Mathematics

What are the fundamental properties of algorithms?

- Formulate and prove theorems:
- Complexity of algorithms.
- Verification of programs.
- Limits of computation.

Theorem as hypothesis; proof as evidence.

Computing as Art

Isn't this neat?

- Building fun systems for their own sake.

This will change the way you think.

- Computers doing things you thought impossible.

"Science in the making" [Bruno Latour]

- Messy process of proposing, arguing for and convincing others of scientific theories.
- Involves emotion, polemic, taste, opinion, politics, skulduggery, …
- Eventually, becomes settled science.
- Hacking matures to technique.

Which View is Correct?

Widely different views strongly held.

- Researchers tend to see just one or two of these views.

In practice, Computing is a mixture.

- Art throws up new ideas.
- Theory cleans them up.
- Engineering turns them into dependable systems
- Science evaluates their scope and limitations.

Exercise: Which Methodologies?

Which methodologies will you use in your research project?

How will they be used?

Will they be combined?

Methodologies in Computing

Computing inherits methodologies from many other disciplines.

Inevitable to use methodology of a discipline to evaluate a computational model in that discipline.

But more than this.

- Due to multi-faceted nature of Computing.

Here are some more examples.

Statistics & Probability

Statistics:

- Used to analyse experimental results.
- In machine learning, e.g., data mining.

Probability: used to model uncertainty:

- In knowledge representation & inference.
- In modelling systems,
  - e.g., stochastic model checking for system biology or computer failure.

Psychology

Cognitive modelling:

- compare model to human/animal behaviour.

Human/computer interaction:

- measure usability of system.

Philosophy

Conceptual analysis:

- Clarify the nature of tricky concepts, such as agency, time, belief, meaning, causality, obligation, etc.

Philosophical logic:

- Represent commonsense knowledge including time, belief, obligation, etc.
- Temporal logics also used in model-checking of computer systems.

Linguistics

Grammars of natural languages:

- Used for speech and written language generation and understanding.

Conceptual analysis:

- of the nature of linguistic utterances, objects and processes.

Sociology

System development:

- To understand how teams of developers interact and are best organised.
- To understand how errors arise.

System deployment:

- To understand the how new technology is received by its intended users.

Biologically Inspired Computation

Brain-inspired computation.

Evolutionary computing.

Ant colonies.

Conclusion

- Computing as exploration of techniques space.
- Importance of hypotheses and their evaluation.
- Dimensions of hypotheses.
- Computing as science, engineering, mathematics and art.
- Computing as the magpie of methodologies.

Exercise

"I can't answer these questions that I've just set."

Exercise

Swap your 1000 word project summary with a neighbour.

Read and critique your neighbour's summary.

Provide feedback to your neighbour and vice versa.