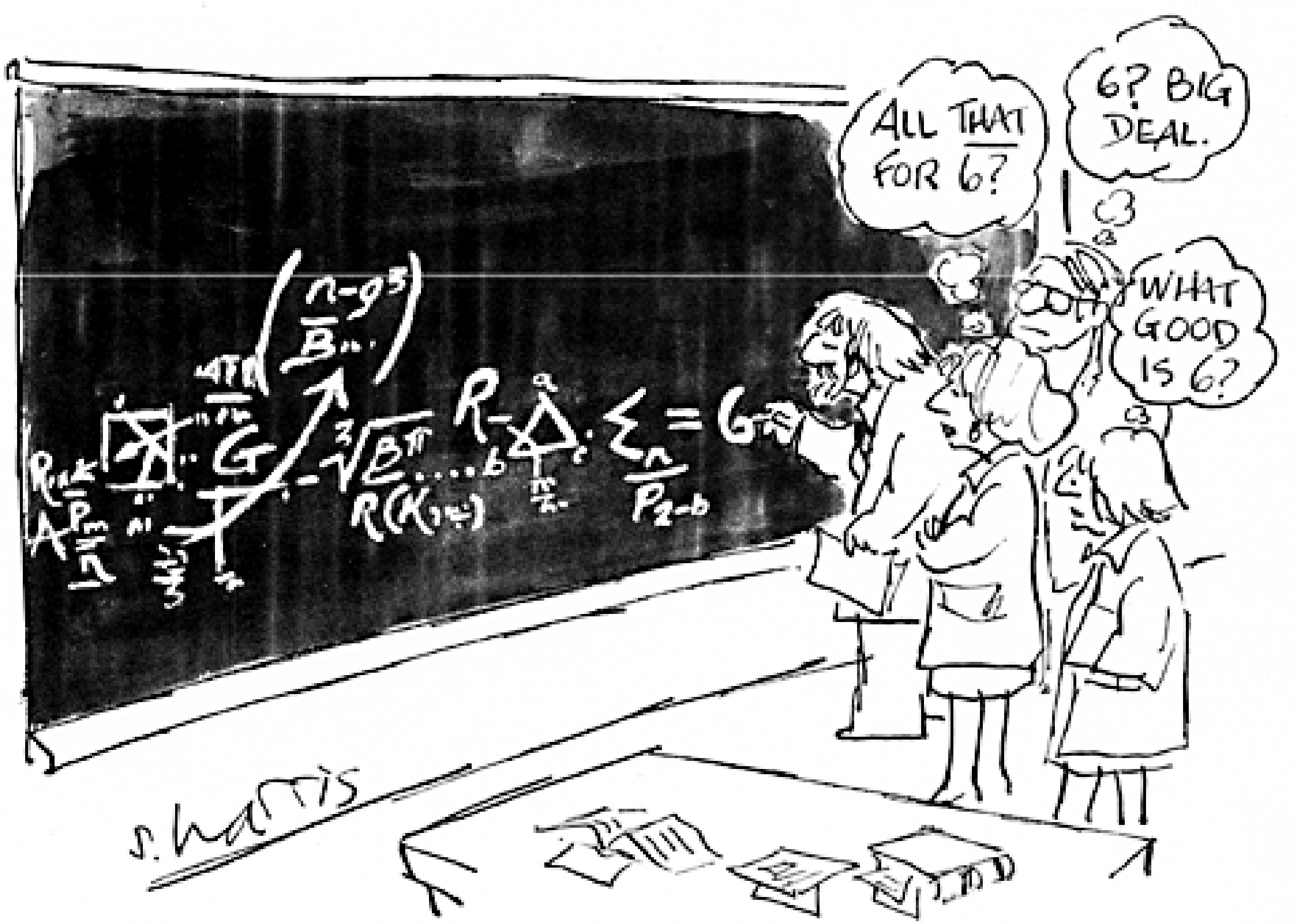
The Need for Hypotheses in Informatics

Alan Bundy
School of informatics
University of Edinburgh

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.
- Informatics should be no exception.
- But the provision of explicit hypotheses in Informatics 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

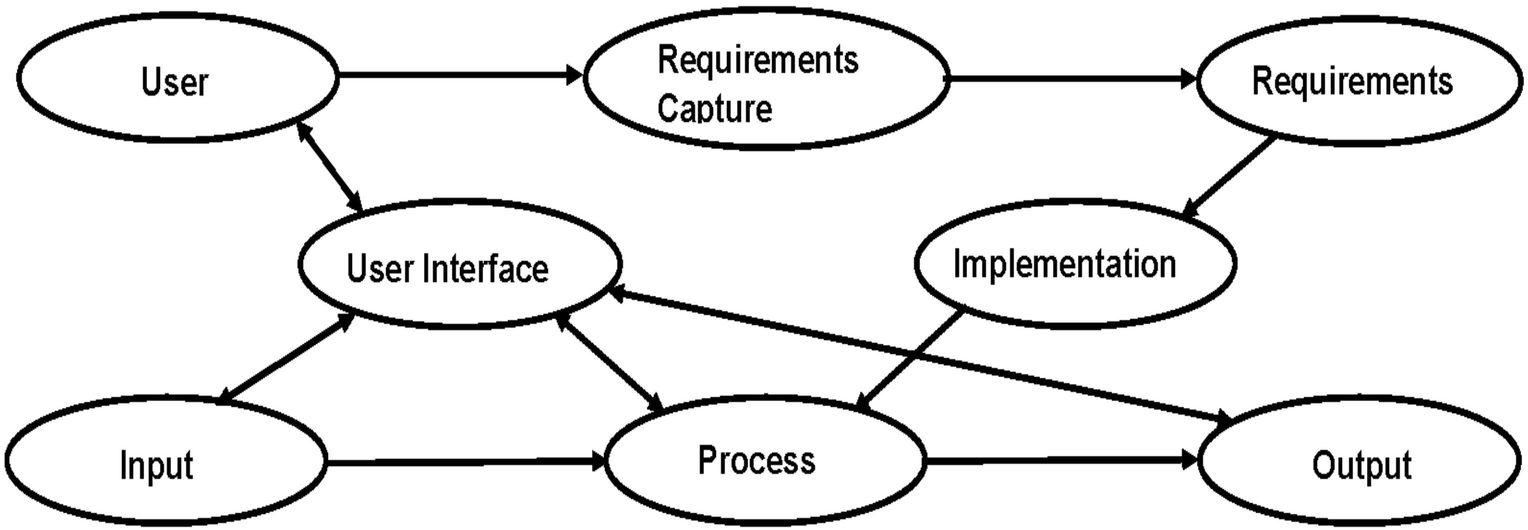
- Invention of new technique,
- Investigation of technique,
  - e.g. discovery of properties of, or relationships between, techniques.
- Extension or improvement of old technique,
- New application of a technique,
  - to artificial or natural systems.
- Combine several techniques into a system.

Hypotheses in 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.
- May be several hypothesis in each publication.

Rarely explicitly stated

Graphical Depiction of Project

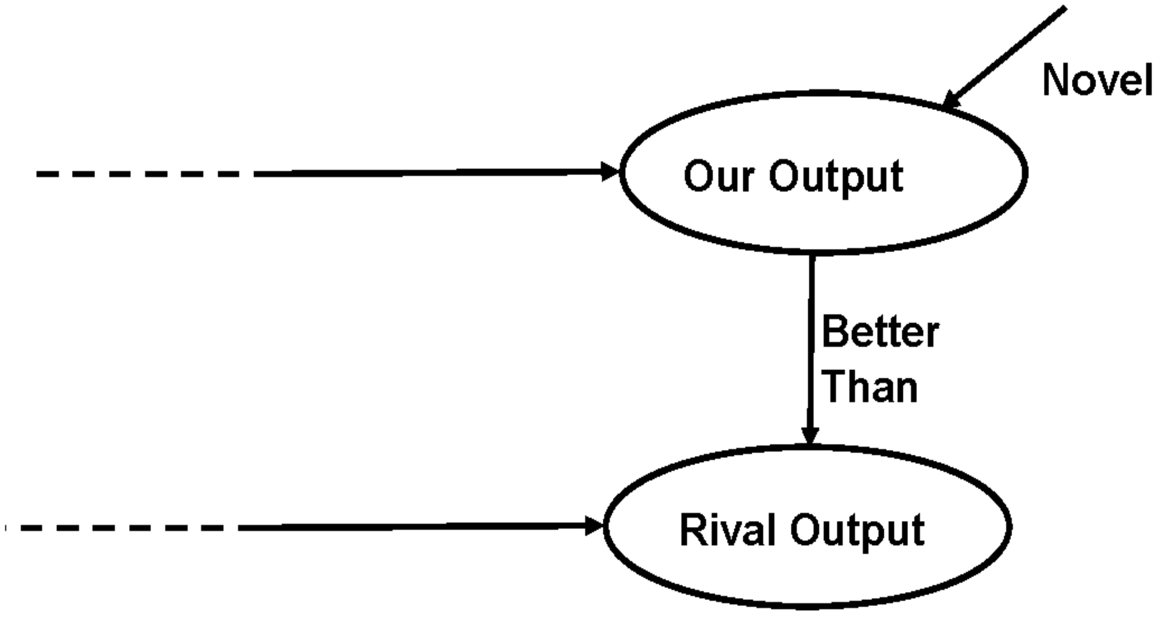


- Systematic generation of hypotheses.
  - By adding novelty label to nodes.

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.

Behavioural Dimension



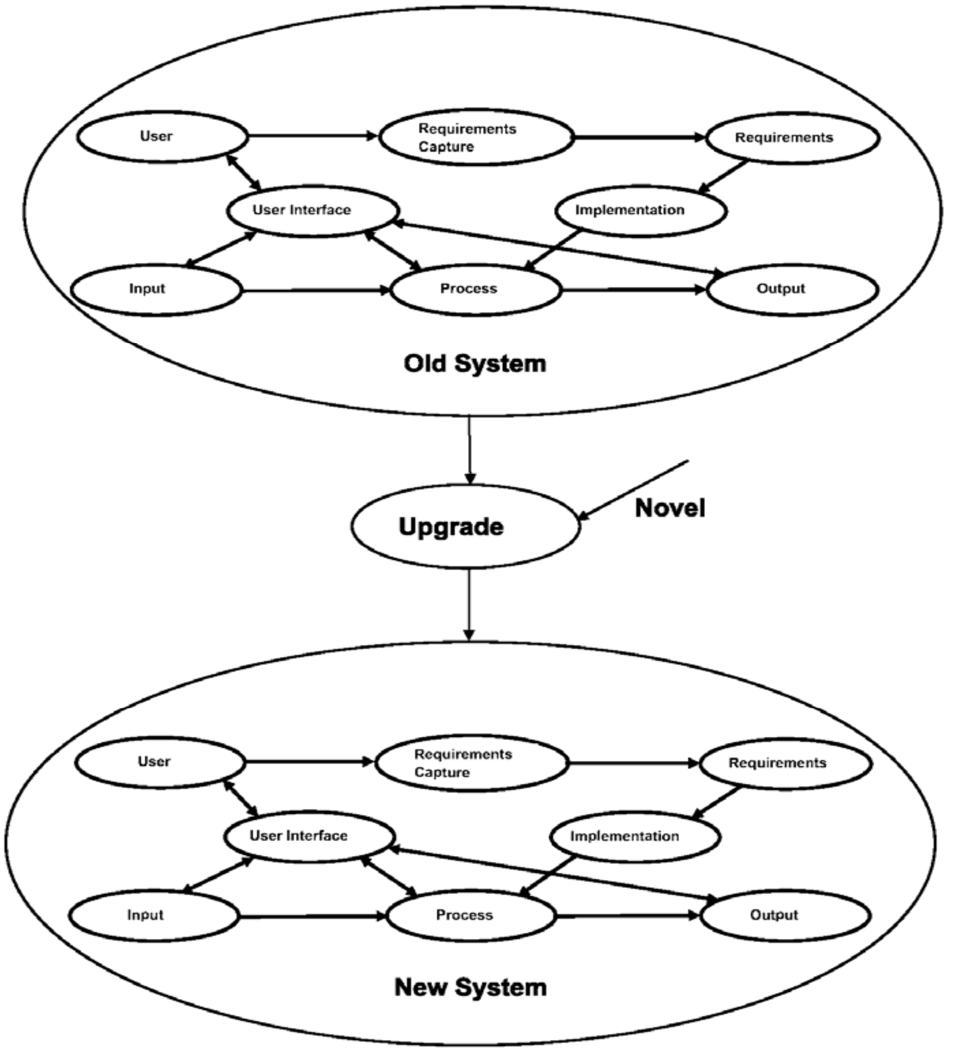
Scientific Dimensions 2

- 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.*

Maintainability Dimension



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

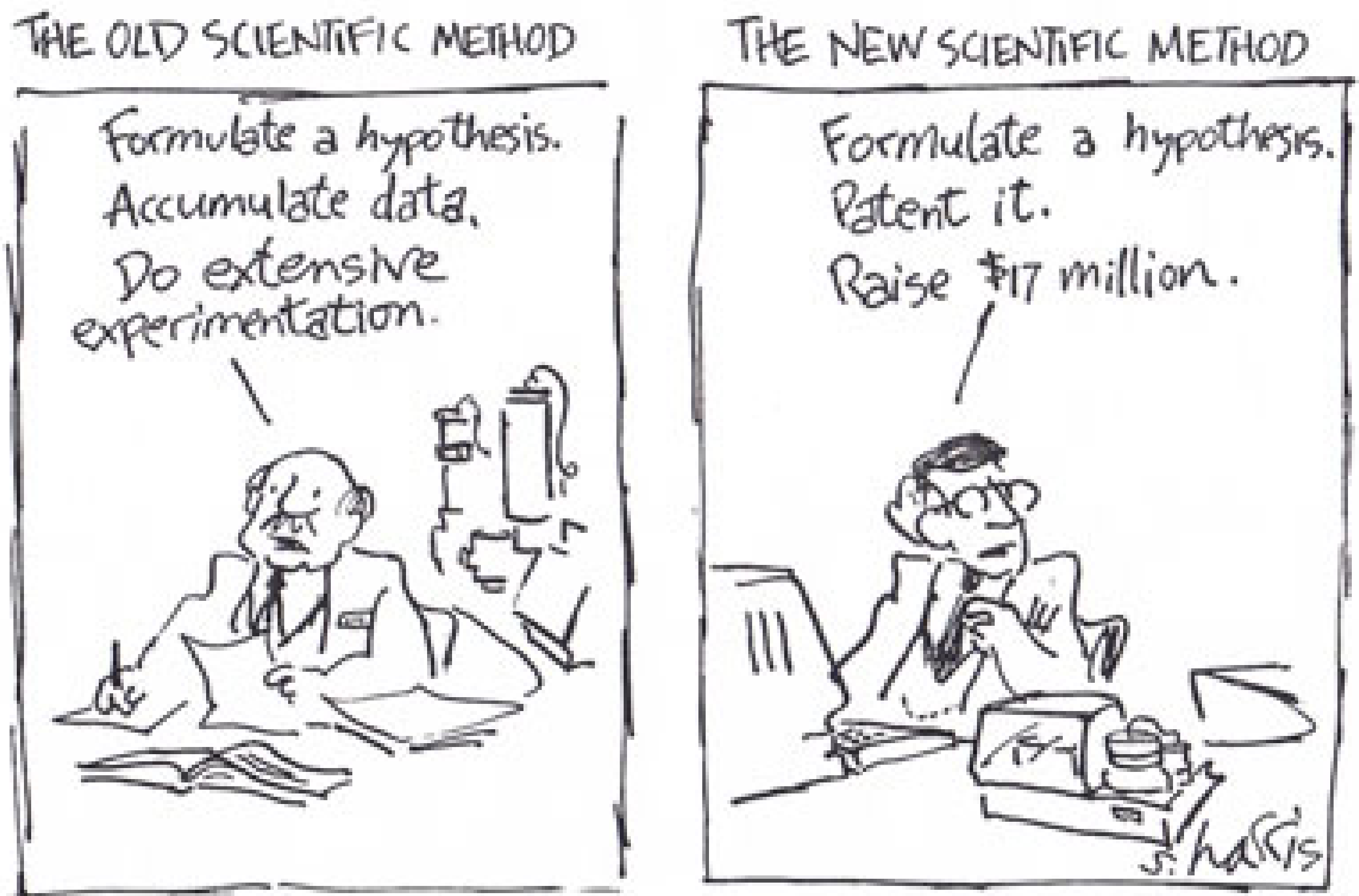
What Informatics hypotheses can you think of?

- Choose system/technique/parameter setting.
- Choose science/engineering/cognitive science 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 tested 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.

Summary

- Informatics advances via formulation of hypotheses,
  - and providing supporting (or refuting) evidence for them.
- Hypothesis typically establish or compare properties along some dimension.
- Property dimensions include:
  - Scientific: behaviour, coverage, efficiency.
  - Engineering: fitness, usability, dependability, maintainability, scalability.
  - Computational modelling: external, internal, adaptability, evolvability.
- Both theory and experiment can provide evidence.