Research Methods
Experimental Computer Science

Edwin Blake
edwin@cs.uct.ac.za

Outline

What is Experimental Computer Science?

Debunking reasons not to experiment

How do you do Experimental Computer Science?

What are Information Artefacts?

In Conclusion

"A science is any discipline in which the fool of this generation can go beyond the point reached by the genius of the last generation."

Max Gluckman

Computer Science

Origins

- Mathematics,
- Engineering, and
- Commercial practice.

Evolved into

- Theoretical,
- Experimental and
- Design (or user) orientated aspects.

Balance and synthesize these aspects

- Mathematics (what is?),
- CS (how to?)

How does this enterprise progress?

Theoretical advances due to new mathematical results

- You prove a theorem
- Fun, short, no argument!
- Not our concern here.

Experimental results

- The real stuff?
- The rest of this lecture …

“Beware of bugs in the above code; I have only proved it correct, not tried it.”

Donald Knuth

“Mathematicians stand on each other’s shoulders while computer scientists stand on each other’s toes.”

R. W. Hamming

What is Experimental Computer Science (ECS)?

ECS is the creation of, or the experimentation with or on, nontrivial hardware and software systems

- These systems, taken broadly, are called computational artefacts.

ECS process:

- Form a hypothesis
- Construct a model and make a prediction
- Design an experiment and collect data
- Analyse results

Why is experimentation so important?

Generally regarded as the basis for the whole scientific and technological revolution that shapes much of our society today.

- Publishing Copernicus's *De Revolutionibus* (1543)
- Galileo pioneered experiments to validate theories (early 17th century)
- Newton *Principia* (1687): laws accord with experimental evidence

Precursors in India, Persia and especially Arab world

- Ibn al-Haytham (Alhazen, 965–1039): shift physics from philosophy to experiment
  - *Optics* (1021): scientific method to investigate vision
  - Famous experiments involved development of the camera obscura to test several hypotheses on light.

Outline

[What is Experimental Computer Science?]

Debunking reasons not to experiment

[How do you do Experimental Computer Science?]

[What are Information Artefacts?]

[In Conclusion]

"when we ignore experimentation and avoid contact with the reality, we hamper progress."

Source: W Tichy, "Should Computer Scientists Experiment More?", *IEEE Computer*, 31(5), May 1998, pp. 32-40

Is Computer Science really an Experimental Science?

Computer Science is “not a science, but a synthetic, an engineering discipline” [Brooks]:

- Phenomena are manufactured
- CS is a type of engineering
- So experimentation is misplaced

But other Sciences:

- Study manufactured entities, e.g., super-heavy elements, lasers
- Make inferences about models, e.g. simulations

Why should we experiment?

Experiments cannot prove anything with absolute certainty

But they are good for:

- Reducing uncertainty about theories, models, and tools
- Leading to new insights and whole new areas of investigation
- Quickly eliminating fruitless approaches

"All truths are easy to understand once they are discovered; the point is to discover them."

Galileo Galilei

Fallacy #1: Traditional scientific method isn't applicable

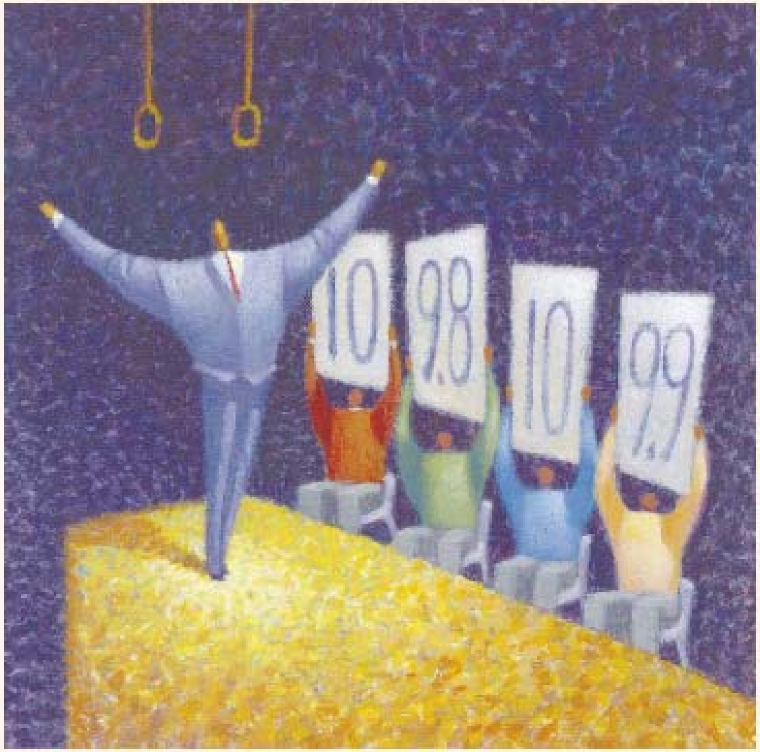
Subject of inquiry is information unlike traditional sciences which study matter or energy

- Example:
  - Object-oriented programming, is it genuinely better?

Rebuttal: To understand information processes, computer scientists must observe phenomena, formulate explanations, and test them. This *is* the scientific method.

Fallacy #2: Current levels of experimentation are enough

- In a study of CS papers requiring empirical backup, 40-50% had no
- Compared to <15% in non-CS papers
- The youth of CS as a discipline is not sufficien justification

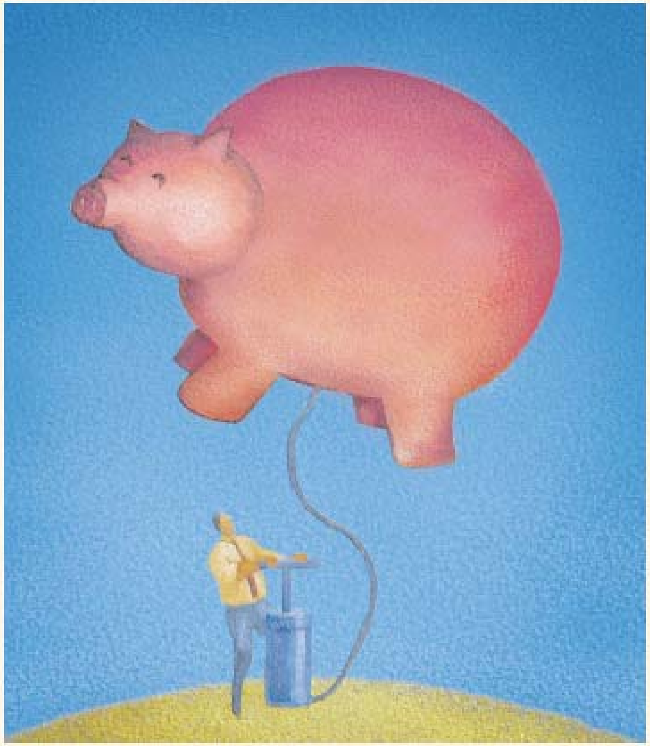


Rebuttal: Relative to other sciences, the data shows that computer scientists validate a smaller percentage of their claims.

Fallacy #3: Experiments cost too much

Experiments can be expensive, but:

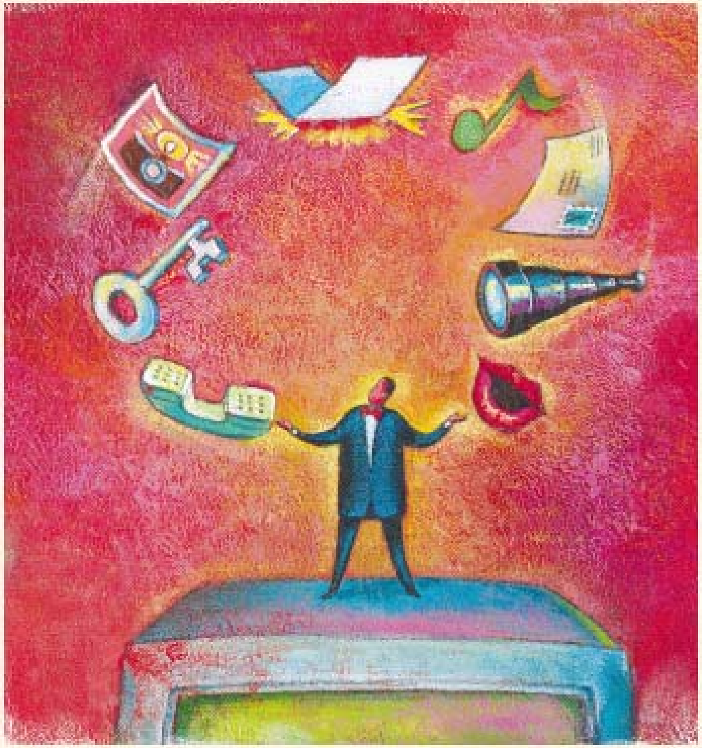
- Often cheaper than the alternative
- The cost may be worthwhile for important questions (general relativity)
- Explore cheaper options (benchmarking)



Rebuttal: Meaningful experiments can fit into small budgets; expensive experiments can be worth more than their cost.

Fallacy #4: Demonstration will suffice

- Demos allow proof of concept and illustrate potential
- But they cannot provide solid evidence



Rebuttal: Demos can provide incentives to study a question further. Too often, however, these demos merely illustrate a potential.

Fallacy #5: There's too much noise in the way

Too many variables, effects swamped by noise

- Answers:
  - Use benchmarks
  - Apply statistical controls from medicine and psychology

Rebuttal: Fortunately, benchmarking can be used to simplify variables and answer questions.

Fallacy #6: Experimentation will slow progress

- Research takes longer → fewer ideas
- Actually weeds out questionable ideas and their offshoots
- Still a place for the hypothesis paper

Rebuttal: Increasing the ratio of papers with meaningful validation has a good chance of actually accelerating progress.

Fallacy #7: Technology changes too fast

“The rate of change in computing is so great that by the time results are confirmed they may no longer be of any relevance” [Mudge]

Look to fundamental long term problems rather

Rebuttal: If a question becomes irrelevant quickly, it is too narrowly defined and not worth spending a lot of effort on.

Fallacy #8: There are substitutes

Theory

- Can be contradicted in practice by incorrect simplifying assumptions

Intuition

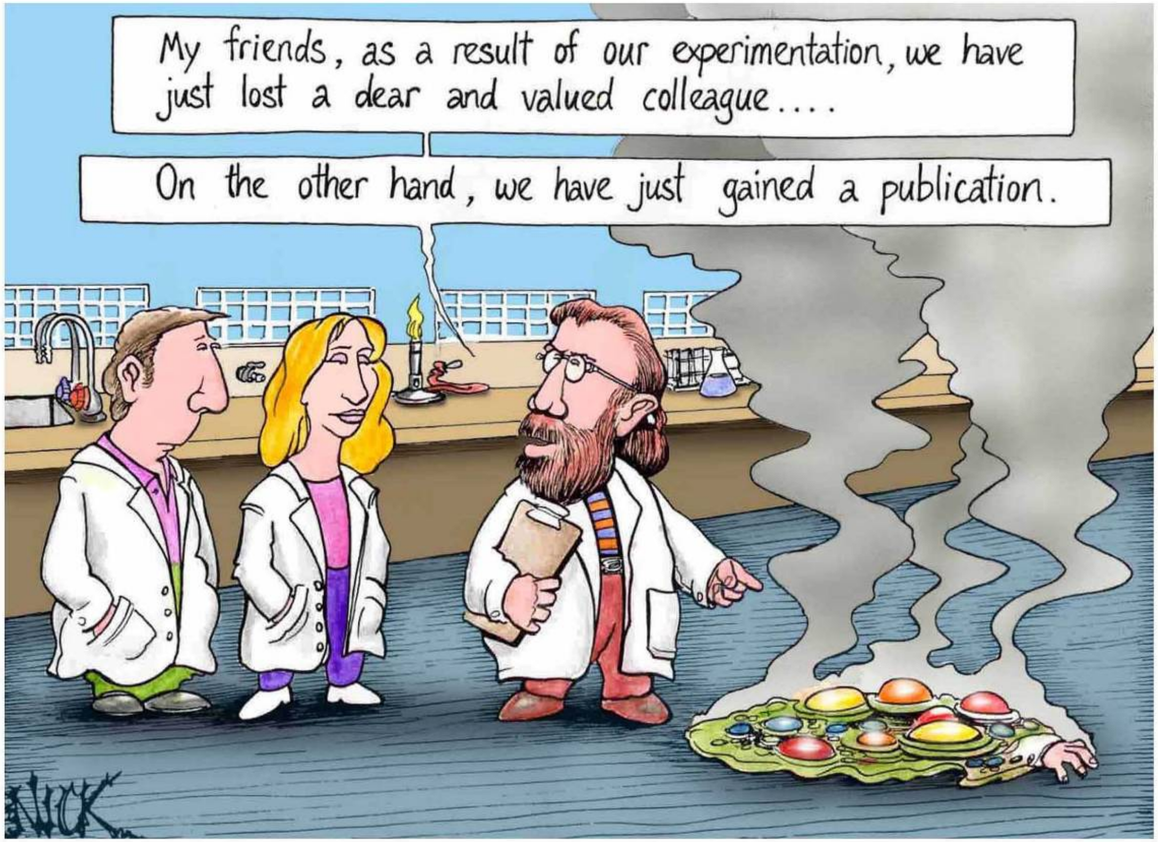
- Fails in the face of counterintuitive results
- E.g., productivity is NOT necessarily improved by typechecking

Experts

- Science must always be backed up by evidence
- E.g., claims about cold fusion

Outline

- What is Experimental Computer Science?
- Debunking reasons not to experiment
- **How do you do Experimental Computer Science?**
- What are information artefacts?
- In conclusion



Experimental Hypothesis

Any scientific research should state a hypothesis, then

- Provide evidence for/against
- Conclude whether it is supported or refuted

Should be:

- Precise, explicit statement
- Falsifiable

Run experiment to confirm or refute

Ensure that the experiment really tests the hypothesis.

"There are two possible outcomes: if the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery."

Enrico Fermi

Elaborating the Hypothesis

Hypothesis might be that system/theory/technique/ parameter P is:

- Good for task X
- Better than rivals Q and R for task X

According to:

- Behaviour — correctness or quality of solution
- Coverage — range of problems to which it applies
- Efficiency — resources consumed

Evidence can be theoretical, experimental or both

- Theoretical evidence — theorem based
- Experimental evidence — testing on a range of examples

Designing an Experiment

Specification needs to be complete and explicit

Make sure the experiment really tests the hypothesis

Requirements:

- Controlled — other factors must be kept constant
- Quantitative — provide numbers
- Coverage — are tests representative of the full range of the hypothesis

Analysing and Reporting Results

Analyse the measured data:

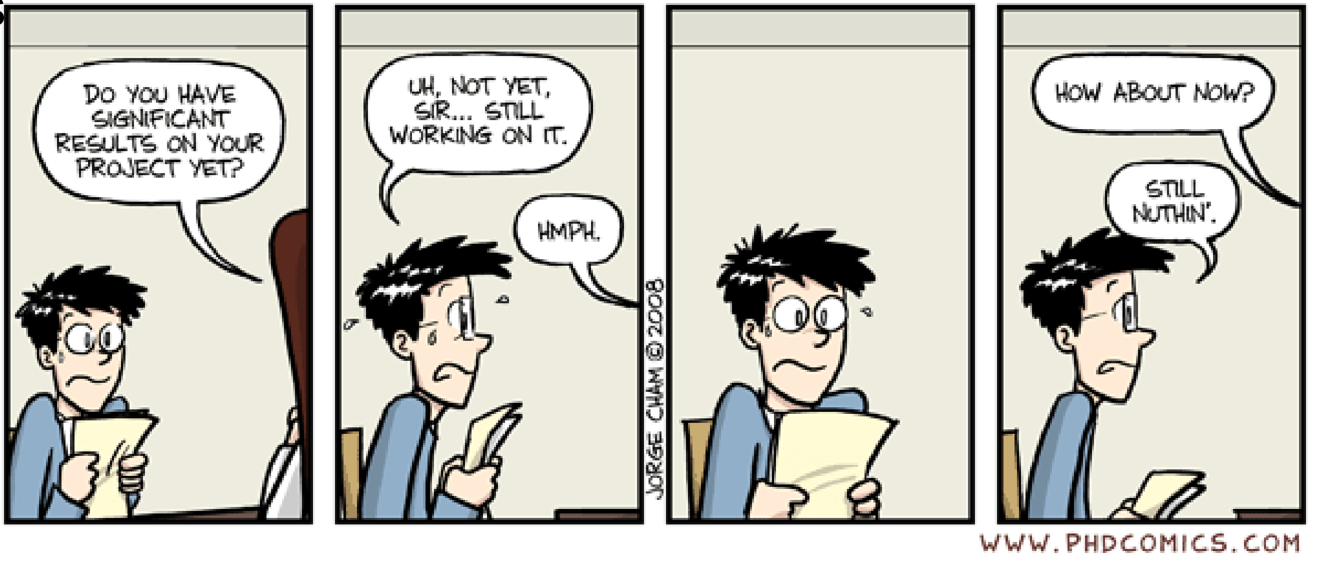
- Does statistical evidence really support (or refute) the hypothesis?
- Make sure differences are not due to chance or natural variability

Be Careful:

- Better to admit to flaws in your methodology
- Don’t generalize without adequate support

Report everything:

- Procedures, results and conclusions
- So that others can replicate the experiment
- And build on your conclusions



Message - Prove your Claims

which means that you have to have claims to prove

and evidence to back you up

and evidence is almost always convincing numbers

from well constructed, all influences considering, set of experiments

that are discussed

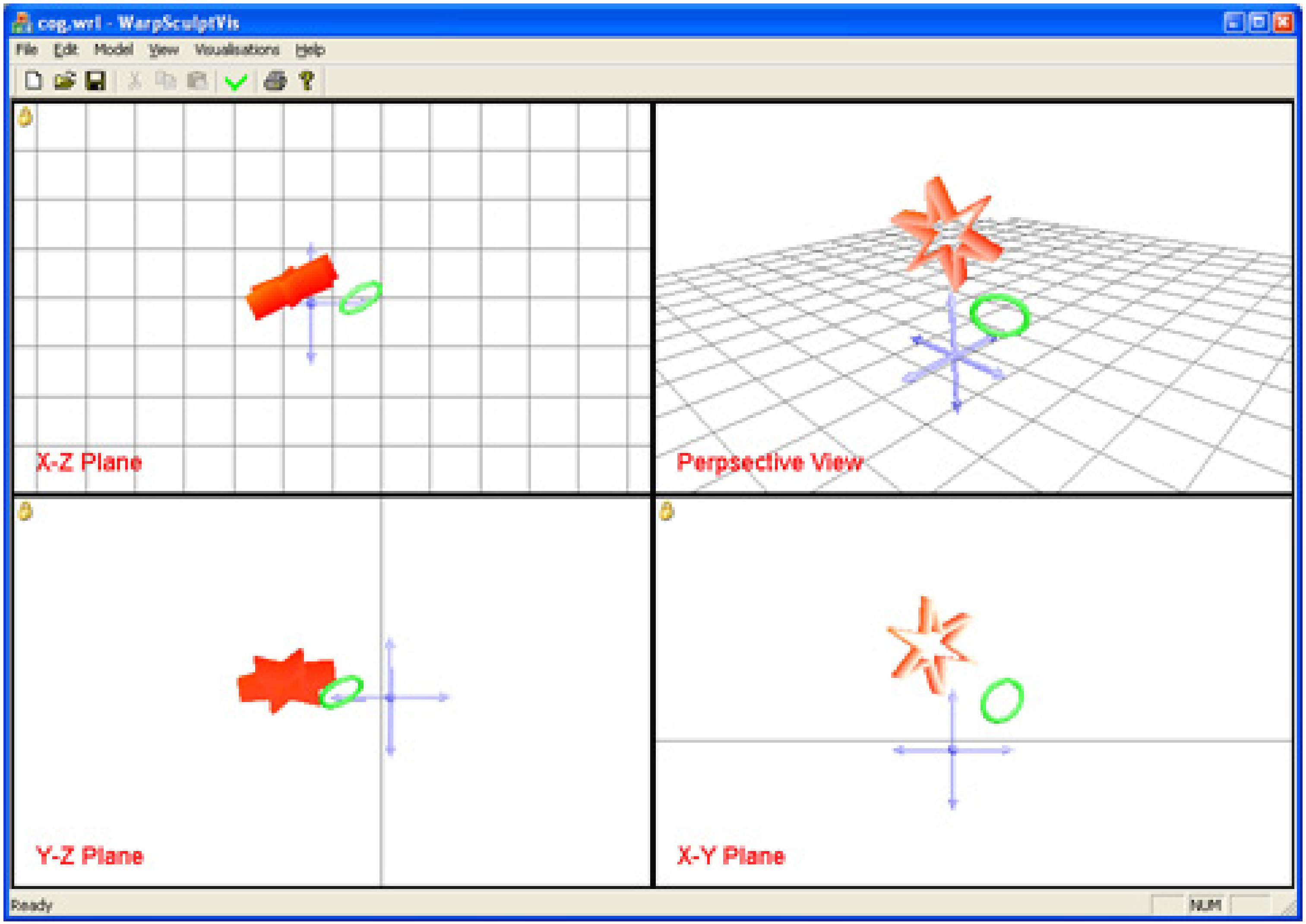
and from which a series of conclusions are drawn

Exercise: Experimental Design

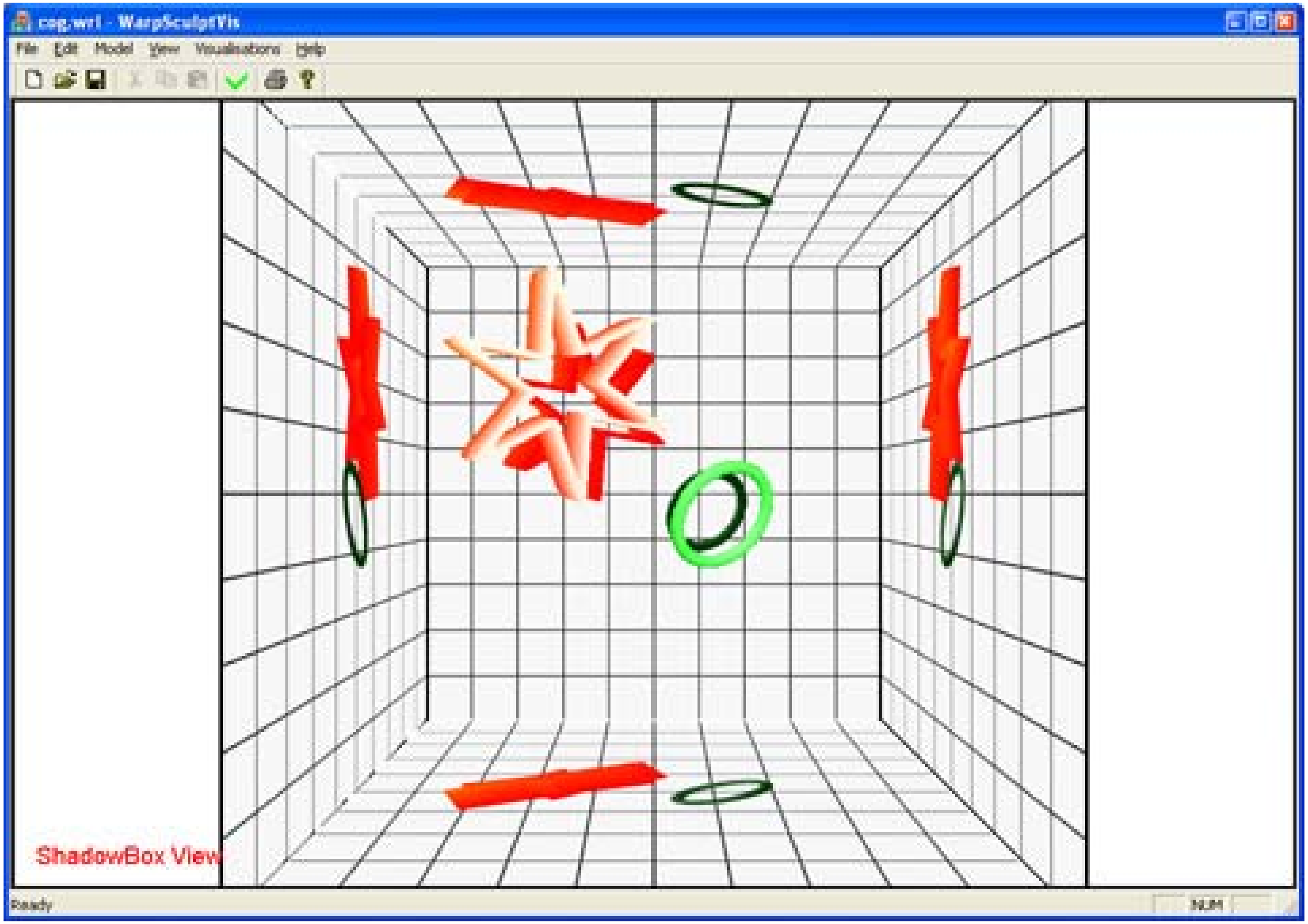
Shadow Box

- Alternative to the conventional layout (perspective + 3 axial orthogonal views) and control (3D cursor) favoured by modelling packages
- A box encloses world coordinate space and models are orthogonally projected like shadows onto its walls
- Box can be rotated
- Shadows can be picked and manipulated

Standard Interface

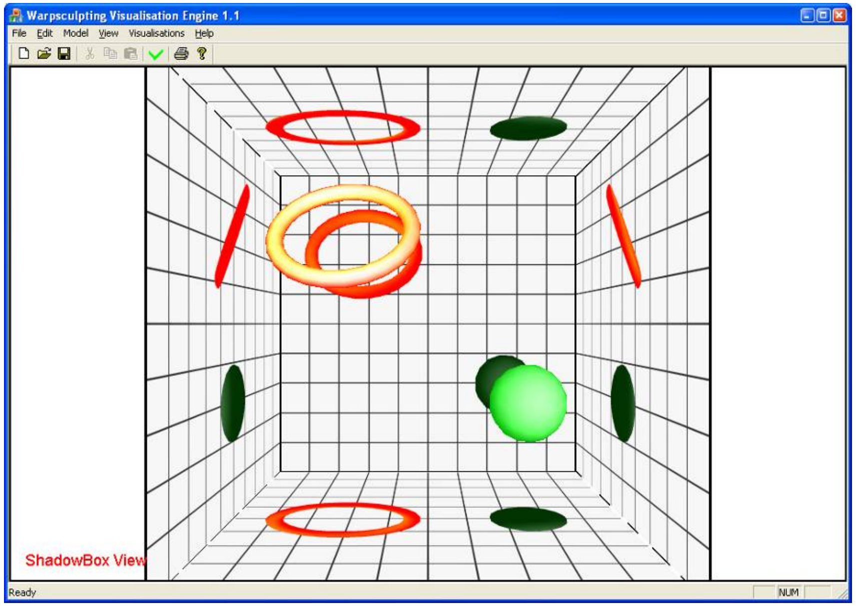


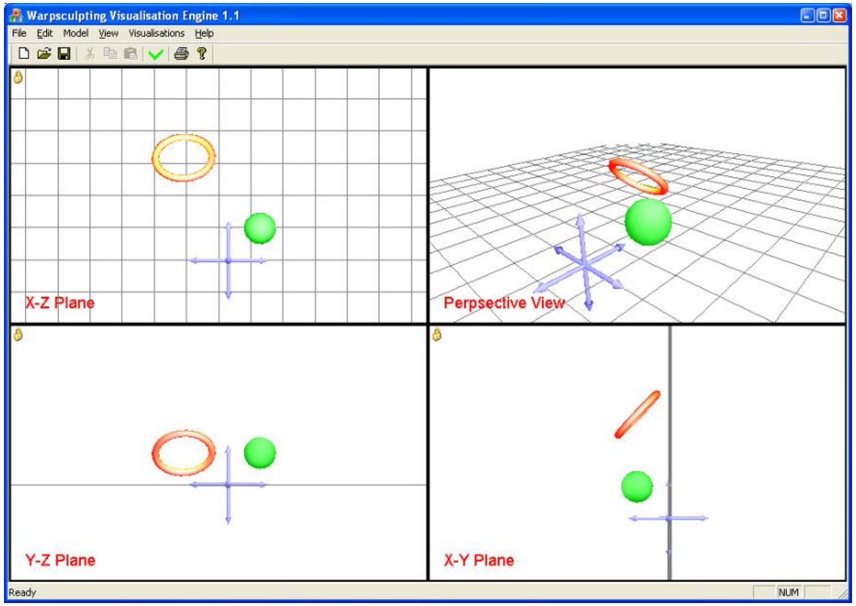
Shadow Box



cog.wrl - WarpSculptVis
File Edit Model View Visualisations Help
ShadowBox View
Ready
NUM

Shadow Box vs. Standard Interface





Hypothesis:

- The Shadow Box allows better control over positioning and orienting 3D objects

- Your task — design the experiment to prove it

Solution: Shadow Box

Setup:

- Two groups,10 in each
- Control group uses Standard Interface
- Experimental group uses Shadow Box

Task 1:

- Pass a ball through a sequence of rings
- Shows — relative movement and positioning

Task 2:

- Rotate a sequence of letters to face the same way
- Shows — relative rotation

Task 3:

- Pass a 3D shape through a sequence of extruded tubes
- Shows — relative rotation and positioning

Outline

What is Experimental Computer Science?

Debunking reasons not to experiment

How do you do Experimental Computer Science?

What are Information Artefacts?

In Conclusion

“Science is built up of facts, as a house is built of stones; but an accumulation of facts is no more a science than a heap of stones is a house.”

Henri Poincaré

Information Artefacts

Synthetic phenomena created by people (not nature)

Artefacts $\equiv$ an instance or implementation of one or more computational phenomena

- Subject of study, and/or
- Apparatus with which to conduct the study

Artefacts are Extraordinarily Complex

- Construction
- Dynamic behaviour

Examples of Artefacts

Hardware systems: computers, chips and circuit boards

Software systems: compilers, editors, expert systems, computer-aided design

Graphic images and animations, robots, hard-to-construct data files (execution traces, Utah Tea Pot)

Programming languages, architectures, protocols, and methodologies (object-orientation, spiral approach, ...)

Why Use Artefacts?

Phenomena overwhelm our ability to understand them by direct analysis

Simply too complex!

And we want to study the dynamic behaviour.

Role of Artefacts: Proof of Performance

Testbed for direct measurement and experimentation

- For quantitative results

Peephole code optimizer

- Conjecture: eliminate redundancies by examining a few generated instructions together

Original RISC prototypes

- Verify performance & implementation advantages

Role of Artefacts: Proof of Concept

Demonstrates that complex assembly can accomplish a particular set of activities

- Geometry engine
  - VLSI leads to low cost 3-D graphics hardware
- Ethernet
  - Feasible to build LAN at low cost with good performance
- Cut-copy-paste
  - Allow useful information manipulation by nonprogrammers via analogy with paper-based text

Role of Artefacts: Proof of Existence

Conveys the essence of entirely new phenomenon (human creativity can produce phenomena never before imagined)

- Mouse
  - Verbal description cannot convey usefulness, needed film of mouse in action

Outline

What is Experimental Computer Science?

Debunking reasons not to experiment

How do you do Experimental Computer Science?

What are Information Artefacts?

In Conclusion

“Your theory is crazy, but it’s not crazy enough to be true.”

Niels Bohr

“If at first the idea is not absurd, then there is no hope for it.”

Albert Einstein

Tips

Keep a note book!

Keep even apparently insignificant aspects of procedure constant between observations

- Time a program, and then repeat the timing while moving the computer's mouse

Data collection and presentation

- Repeat observations several times
- Raw data must be included in lab reports

Data analysis requires (some/lots) of statistics

Testbed

For experimental measurements of systems or algorithms

- embed your system in a testbed
- record parameters and settings as well as timings and output
- See Ramage & Oliner, 2007, “RA: ResearchAssistant for the computational sciences”: doi.acm.org/10.1145/1281700.1281719

Reproducible Research: Beyond Experimental Research

RR is the idea that in computational sciences, the ultimate product is not a published paper but rather the entire environment used to produce the results in the paper (data, software, etc.)

Claerbout: a scientific article is merely advertisement of scholarship; the real scholarship includes software and data which went into producing the article.

- See IEEE "Computing in Science & Engineering" Jan 2009: Special Issue on Reproducible Research. doi.ieeecomputersociety.org/10.1109/MCSE.2009.14
- Kovacevic: "How to Encourage and Publish Reproducible Research" www.andrew.cmu.edu/user/jelenak/Repository/07_04_Icassp_Kovacevic.pdf
- reproducibleresearch.net

How to make a paper reproducible? I

First: A paper with the theory, algorithm, or experiments.

- A block diagram or a pseudo-code description can do miracles!

Then a web page containing the following:

1. Title
2. Authors (with links to the authors' websites)
3. Abstract
4. Full citation of your paper, current publication status, + PDF
5. All code to reproduce all the results, images and tables.
   - Code well documented; a readme on how to execute it
6. All data to reproduce all results, images & tables.
   - Add a readme file explaining what the data represent
7. Configurations on which code was tested (software version, platform)
8. An e-mail address for comments, remarks and bug reports

How to make a paper reproducible? II

Depending on the field, it can also be interesting to add the following (optional) information to the web page:

1. Images (add their captions, so that people know what Figure xx is about)
2. References (with abstracts)

For every link to a file, add its size between brackets.

- Allows people to skip large downloads.

- See Reproducible Research Repository (rr.epfl.ch) for a list of reproducible papers.

Conclusion

ECS is a fundamental underpinning of the information age

Synthetic: studies phenomena that are entirely the product of human creation.

Information artefacts are extremely complex and can only be understood via empirical observation

Complexity often precludes direct theoretical analysis

Your Project (almost certainly) **Needs ECS**

References

Walter Tichy, "Should Computer Scientists Experiment More?", IEEE Computer, 31(5), May 1998, pp. 32-40

- ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=675631

Alan Bundy's "How-To Guides"

- homepages.inf.ed.ac.uk/bundy/how-tos/how-tos.html
- A Scientific Checklist - a list of some useful questions you should ask about your own research
- The Researchers Bible - a guide to getting a PhD in AI

Doug Baldwin, "Using Scientific Experiments in Early CS Laboratories"

- doi.acm.org/10.1145/134510.134532

Dror Feitelson, Experimental Computer Science: The Need for a Cultural Change. Manuscript, 2005.

- www.cs.huji.ac.il/~feit/papers/exp05.pdf

Some Examples

The following papers are examples of experimental computer science with an emphasis on methodology that may be of general use:

- Stallmann & Brglez, 2007, High-contrast algorithm behavior: observation, hypothesis, and experimental design. doi.acm.org/10.1145/1281700.1281712
- Dinda et al, 2007, The User In Experimental Computer Systems Research. doi.acm.org/10.1145/1281700.1281710