Research Proposal

When asked which career achievement I value most, I always put in first place the fifty-two grad students I have supervised. Each student participates in all three stages of research, refining the topic into a well-defined problem, solving the problem, and writing the results as a thesis. I think that the first stage is the most important and that student initiates topic based on their owninterests, not on mine. Thus, each student works on a topic he or she chooses, which is essential because good research requires a passion for its topic.

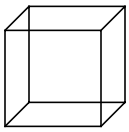
My passion is the empirical research methodology I teach my students. It starts with a simple abstract model, which the student first creates then explores doing casual experimentats. The student then refines the model based on the results and the refined model is further explored. The process ends with a model that describes the phenomenon. Sometimes the model provides the abstraction for a formal experiment, sometimes for an implementation, sometimes for a theorem. According to the students, and to their employers, it is a successful method of graduate training.

More generally, my passion is how we learn by empirical exploration, and by all types of experimentation. The research proposed in this application is a response to my passion. Until now, my engagement with experimental methods has been *ad hoc*: as a problem revealed itself the student developed a solution specific to it. So to say, I was myself engaged in exploratory experimentation. Now I wish to go on to the next stage, formalizing and codifying a career's insights in a monograph to expose them to a wider audience. Most of my experimental experience is in computer graphics (CG) and human-computer interaction (HCI). I have also taught courses on simulation and experimentation for performance analysis, and supervised an experiment based on an AI simulation [6].

A. Introduction

Exploratory experimentation is natural and ubiquitous: anthropologists say it is rooted in everyday life [4]; psychologists say it is the base of infant learning [8]. All sciences use it, even mathematics: Gauss, for example, claimed he discovered mathematical truths 'through systematic experimentation'. But, I must teach it to my students. Computer science education emphasizes thought, planning and formalization, and students are wary just trying something out, at least in front of their supervisor. In contrast, my contacts in industry aver that where commercial software innovates it is rife with ill-understood constructs that just happen to work: innovation precedes understanding, as exploratory experiments precede theory.

Exploratory experimentation is well understood because there's not much to understand. It emphasizes destination over route; its methodology is 'anything goes' [5].



The same cannot be said for formal experimentation. Exploratory experiments are for finding truth. They are rarely convincing to a third party. (The bistability of the Necker cube shown to the right, which is immediately apparent, is an exception.) Formal experimentation is for sharing truth. Conducting a formal experiment, *according to the practices of a scientific community*, is the requirement for making public an empirical claim. Practices are specific to a discipline, and are slow to mature. In the eighteenth century, for example, a knowledgeable, disinterested third party had to witness an experiment in person. Experiments were displayed before an audience before they were published. Now the standard is replication: a competent researcher should be able to redo the experiment and get the same result. (Until recently referees were expected to replicate before accepting a paper for publication.) To become 'competent' a student of experimental physics must learn a complex network of standards and procedures that completes the published descriptions of experiments.

Computer science is a young science, six or seven decades old. It has few stable experimental practices; my experience is that few experiments are replicable. (See §B for details.) We need standard practices, and we need them now, not a century from now. Formal experiments are common in algorithm analysis [10], in performance measuremeny and in artificial intelligence, not to mention CG and HCI.

Formal experiments are widespread because computers are physical objects of extreme complexity, and because the human mind best tames complexity using the model/experiment method. Knowledge gained using like this sometimes reduced to theorems, but more often it is the outcome of a formal experiment.

In the past computer science has advanced by finding in other disciplines concepts and processes that improve its own practices. I believe that it can do so to improve experimental practice. In the next section I discuss experimentation in physics, the most obvious discipline from which to borrow. While physics has much to offer there are important differences between the demands of physics

and those of computer science. In the following section I examine the social sciences and find further ideas there.

B. Experimentation in Physics

This section introduces three prominent concepts of experimental physics: cumulation, replication and isolation. Each concept is defined as understood in experimental physics. Instances of it in computer science, largely of my students, are described, and research ideas I plan to explore.

B.1. Cumulation

In all aspects of all sciences new research should be cumulative. Its goal is not to stand by itself, a result independent of all other research, but to be part of a whole that is the collective knowledge of the discipline. If pieces of research are to cumulate, they must share concepts and processes. Researchers must form a community based on common ideas and values. For example, we judge the health of a experimental discipline not by the exploits of a few virtuoso experimentalists, but by the ability of the community to steadily accumulate a coherent body of knowledge.

There exist many examples of cumulation in computer science experimentation, such as experiments on Fitts' Law in HCI [15] or measurements of network traffic [17]. But far too many experiments are one-off evaluations or comparisons. Huo assembled all experiments on visual query interfaces, finding only two that were similar enough to compare, which produced opposing results. Better to understand cumulation in computer science I plan to assemble as many examples as possible of successful cumulation, to see what they have in common. Preliminary work suggests that they share strong models that provide extensive abstraction. Much of each model originates outside computer science, suggesting that cumulation requires a multi-disciplinary approach.

B.2. Replication

Replication is a necessary condition for cumulation, though not a sufficient one. When it is possible to do the same experiment and obtain the same results the experiment is said to be replicable. Ideally, another experimentalist needs only the published paper plus knowledge of the experimental practice of the discipline to perform a successful replication. The increasing complexity of modern experimentation puts a strong emphasis on the second requirement, Students of physics get exposure even as undergraduates, where they replicate historical experiments in lab courses. As graduate students they are expected to replicate existing results in their area before undertaking their own experiments. New apparatus or modifications of existing apparatus are extensively tested by replicating previous results before being put into service.

As a young faculty member I naïvely expected students to replicate existing results. It was rarely possible. Experimental tradition was so weak that no paper could contain all the details needed to replicate its results. Later over several terms I had students in a user interface course replicate classic results in HCI as assignments. Some experiments replicated consistently, despite variations in implementation and context; other replicated not at all.

Replicability is essential. It is difficult because it is the property of a community, not of a single experiment or its write-up. To investigate replicability I will hire undergraduate students to reimplement and run classic experiments, carefully recording every assumption made in order to complete a successful implementation. Comparing assumptions will give an estimate of community practice. Comparing implementations will, I hope, lead to a useful proxy for implementation, so that its doesn't have to be treated as a random variable in the analysis.

B.3. Isolation

Another practice of experimental physics is the creation of special environments in which the effect to be measured dominates the behaviour of the system, which greatly simplifies the models supporting the experiment. Newton, for example, did not expect his laws of motion and gravitation to predict the motion of a leaf falling from a tree. The model required to make sense of the measurements was truly impossible to create, as it remains to be. As an alternative to leaves, corrupted components of the sublunar sphere, he had the uncorrupted planets, which provided observations, for more on which see §C.2, in which forces other than gravity play a negligible role. Isolation, as practised in the twenty-first century, now rests on, an immense cumulation of well-tested models and experiments. Current experimenters do not expect to get replicable results from corrupted specimens, for reasons that are well-supported in current physics.

Thus, replication is at odds with the desire in computer science for ecological validity, which usually amounts to doing experiment in conditions that are as 'real world' as possible. Like the real world falling leaf, real world examples in computer science are too complex to admit understandable models. Realising that there are perfectly valid reasons for rejecting replicability, we should not expect such experiments to be replicable, and

ipso facto neither should we expect them to cumulate.

Isolation is practised in computer science, the experiments of Mytkowicz et al. [16] is an example of systematically removal of confounding factors that would be impressive in any experimental science. Preferring a generality over ecological validity I have tried to guide my students to experimental designs with as few confounding features as possible.

Ph.D. student Jiwen Huo [9] introduced experimental manipulations that isolated perceiving from truth calculation in the minds of subjects evaluating the truth of Boolean queries. Systematic differences between the isolated variables when the displays were images or text allowed us to conclude that there are two qualitatively different ways for comprehending Boolean expressions. Isolation plus a strong model enabled a result that has wide generality.

M.Math student Ed Dengler [2], using a questionnaire, showed that the same graph, differently laid out, can produce very different interpretations. Despite deliberate use of a heterogeneous subject group the data was very clean and the inter-subject results consistent, owing to a six month period of pilot experiments during which confounding variables were gradually stripped from the graphs presented in the questionnaire.

These three examples show that isolation is an effective experimental strategy in computer science. The next step is to understand the trade-off between real world and isolation in computer science experimentation. **<<More, including Kroeger if there is space>>**

c. Experimentation in the Social Sciences

The previous section described possible research based on analogies with physics, which of all science has the most mature experimental practices. Computer science, however, cannot merely imitate physics, but should create its own community standards. In this section we examine two social science disciplines, psychology and economics, that have immature experimental practices that offer some ideas that computer science could adopt with benefit.

c.1. Standardization and Operational Definition

Models are most effective when they are based on a deep reservoir of standards. Standards have two faces: one face is a model component, an abstract black box with well-defined properties; the other face is an experimental practice used to control. For example, in an experiment that depends on vision, the physical display, CRT, LCD or OLED, appears in the model as a device capable of putting optical patterns on the retina of the subject. It has properties like spatial and temporal resolution, colour gamut, surface reflection and so on. The same device. It is also a physical devices that is part of the experiment. To control resolution, colour, reflection et al., there are experimental practices that focus the pixels, that calibrate the colour, that adjust room lighting, et al. The experimenter is expected to decide whether or not these aspects of the display are relevant to the experiment, and if they are, how to control them.

This example is easy to understand: an easy to identify object has relevant physical properties that are standardized by physical manipulation. Most experimentation in physics requires exactly such standardization. In psychology things are more complex because experimenters wish to use model elements, like attention, or cross-hand interference, that are ill-defined fuzzy and vague. Psychologists use operational definition to solve such problems. Cross-hand interference, for example, is defined to be zero when the performance of a task by one hand is uninfluenced by independent tasks performed by the other hand. The model element is defined, not in terms of its intrinsic relationships with other elements, but by its experimental consequences, which makes it easy to integrate into an experiment. Now that experiments on cross-hand interference can be performed, their results will, one hopes, allow the definition to evolve, accreting conceptual baggage until a well-defined psychological entity emerges. Thus, operational definitions have allowed psychology to pass more rapidly toward well-defined theoretical concepts than physics did.

My earliest research in computer graphics enabled the adoption of colour standards by the computer graphics community [1], closely followed by using colour standards to perform experiments in what is now HCI [19]. Later, M.Math. student Raymond Yiu [21] and Ph.D. student Martin Talbot [20] used operational definitions to interact with scorers in double-blind experiments. Huo also used operational definitions of perceive and comprehend in her experiment [9].

<<More>>

c.2. Observation

Many scientists, astronomers for example, are unable to perform controlled experiments. Instead, they observe phenomena they cannot control and look for cross-observation correlations among observed properties and use them to refine models. Astronomy, of course, has strong models of the four elementary forces, which constrains models and gives them substance. Without a strong

uncontested theory for establishing models, economics, a science in which most experiments would be unethical, has developed econometrics, a collection of strong methods for drawing conclusions from noisy and confounded observations.

During a time when my students had access to managers and programmers at the IBM Toronto laboratory they did observational studies investigating software engineering [3, 7, 13]. The observations were dictated by intuitive models, which were then more or less confirmed when the observations showed trends similar to those predicted by the models. These data were, however, not sufficiently extensive to benefit from econometric methods. Observational experiments I intend to pursue, which will benefit from econometric techniques are the longitudinal experiments described below (§C.3).

Equally interesting as an object of study is the tension between ecological validity, close imitation of real world instances, and the generality sought by science (§B.3). Suppose we accept a pessimistic interpretation of Mytkowicz et al. [16]. Then performance measurements that are both replicable and general are possible only by stripping away very fundamental features of real computers, such as CPU instruction and data caches. We can then progressively re-enable the stripped away features, and get a sequence of measurements that are correspondingly more noisy. Deterministic replication gives way to statistical replication, with 'suitable' sampling from a poorly defined universe of system configurations and loads. At the end of this continuum are real, non-instrumented systems processing real loads in the real world. Such systems can only be observed, controlled experiments are not possible. All the power of econometrics will be needed to make the sequence of measurements and observations comparable. I intend to do this experiment which, in addition to its intrinsic value, will be a real challenge to the ideas in this proposal.

C.3. Longitudinal Experimentation

Consider a biological experiment that wishes to compare the knees of twenty year olds to the knees of fifty year olds. The simplest way of doing so is to compare a young group to an old group. This procedure has both systematic artifacts – the old group was twenty years old thirty years ago, when nutrition and excise habits were quite different – and random artifacts – knees actually differ a lot among the population. Medical statistics has created many methods for reducing the variation, such as choosing a sample of matched pairs. All would agree, however, that the preferred way of doing this experiment is to choose one group of twenty year olds, measure them, and wait for thirty years, if, of course, you can stand the wait. The constraints of longitudinal experimentation – subjects have to get on with their lives – means that the data is largely observational.

Longitudinal studies are rare in computer science: last year's system, interface, algorithm is implicitly taken as contributing only to the history of technology. As a result, where longitudinal experiments exist, such as those of M.Math students Celine Latulipe [12] and Erin Lester [14], they examine short term changes in users or in adaptive systems such as JIT compilers. The two studies I supervised, both controlled experiments, were not successful: to observe learning, thirty hours of use is too short and a dozen subjects is not enough. These results suggest that longer observational studies are best: the Schroeder & Gibson [18] study of hard disk errors is exemplary, but depended on the existence of just the right dataset.

Creation of datasets, which are used and reused, is usually hard work. But without them longitudinal analysis of observations. Recognizing this, economists regard creation of a publically available dataset as an important research achievement. Several years ago, to present at a seminar, I used system logs to make observations showing a contagion model of software adoption. I intend to create longitudinal observational datasets containing feature sets of 2D and 3D graphics software to identify clusters of features that appear together. The will be compared to the results of pilot experiments done by undergraduate Eoghan Sherry, which will be continued. They modelled user interfaces as group algebras. The subgroup structure is created by clusters of features that require each other to be useful.

D. Summary

The emphasis of the proposed research is improving the state of experimentation in computer science. Other disciplines have taken centuries to create mature experimental practices: I propose that by explicitly adapting experimental techniques from other disciplines, as my students have been doing implicitly for two decades, we can speed the maturing of experimentation within computer science.

During the five years covered by this application I expect to graduate a further dozen students, most of whom will go to industry. They will choose their own research topics, but all will learn my model/explore methodology and many will design, pilot, run and analyse formal experiments. Their results will be published in appropriate venues. The projects explicitly proposed in this appli-

cation I will do myself, assisted by undergraduate students with URAs.

While this is going on I will complete a research monograph that I have begin writing, and of which this application is a brief synopsis. The monograph will, I hope, disseminate these experimental techniques throughout computer science, as Johnson did for theoreticians [10]. Doing experiments that cumulate is exacting, and I hope that my research will raise the status of experimentation, and contribute to the formation of an experimental community with shared goals and standards.

E. References

My students' names are in bold face.

1. Cowan, W. 'An Inexpensive Method for Colour CRT Calibration', *Proceedings of SIGGRAPH'83*, 130–137, 1983.
2. **E. Dengler** & W. Cowan, W., Human perception of laid-out graphs. In: S. Whitesides, (Ed.), *Proceedings of Graph Drawing Symposium 1998, Lecture Notes in Computer Science 1547*, New York: Springer, 441–443, 1998.
3. **d'Oliveira, E.** *Growing Software: An Economic Analysis.* M.Math. Essay, University of Waterloo, 1996.
4. Dunbar, R. *The Trouble with Science.* London: Faber and Faber, 1995.
5. Feyerabend, P. *Against method: outline of an anarchistic theory of knowledge.* London: Verso, 1975.
6. **Fourquet, E.**, Larson, K. & Cowan, W. 'A Reputation Mechanism for Layered Communities. *SIGecom Exchanges*, 6(1): 11–22, 2006.
7. **Glover, T**. *Education Strategies for Optimized Output in the Presence of Skill Obsolescence.* M.Math. Essay, University of Waterloo, 1999.
8. Gopnick, A. M., Kuhl, P. K., & Meltzoff, A. N. *The Scientist in the Crib: Minds, Brains, and How Children Learn*. New York: Morrow, 1999.
9. **Huo, J.** & Cowan, W. 'Comprehending Boolean Queries'. *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*, 179–186, 2008.
10. Johnson, D. *A theoretician's guide to the experimental analysis of algorithms*. Manuscript of invited talk presented at *AAAI-96*, Portland, OR. Accessed at www2.research.att.com/~dsj/papers/experguide.pdf on 28/9/2010.
11. Latour, B. *Science in Action: How to Follow Scientists and Engineers through Society.* Cambridge: Harvard University Press, 1987.
12. **Latulipe, C.** *A Longitudinal Target Selection Study with Force Feedback*. M.Math. Essay, University of Waterloo, 1999.
13. **Lau, T.** *Cost-benefit Analysis in Software Engineering.* M.Math. Essay, University of Waterloo, 1992.
14. **Lester, E.** *Early Language Learning is a Good Model for Studying Early Interface Learning.* M.Math. Thesis, University of Waterloo, 2005.
15. MacKenzie, I. S. 'Motor behaviour models for human-computer interaction'. In J. Carroll (Ed.) *HCI models, theories, and frameworks: Toward a multidisciplinary science.* San Francisco: Morgan Kaufmann, 27-54, 2003.
16. Mytkowicz, T., Diwan, A., Hauswirth, M., Sweeney, P. 'The Effect of Omitted-Variable Bias on the Evaluation of Compiler Optimizations'. *IEEE Computer*, 43(9): 62-67, 2010.
17. Paxson, V. & Floyd, S. 'Wide Area Traffic, the Failure of Poisson Modelling'. *IEEE Transactions on Networking*, 3: 226–244, 1995.
18. Schroeder, B. & Gibson, G. 'Disk failures in the real world'. FAST'07: 5th USENIX Conference on File and Storage Technologies, unpaginated, 2007.
19. **Schwarz, M.**, Cowan, W. & Beatty, J. 'An Experimental Comparison of RGB, YIQ, LAB, HSV and Opponent Colour Models'. *ACM Transaction on Graphics,* 6: 123–158, 1987.
20. **Talbot, M.** *Spatial Auditory Maps for Blind Travellers.* Ph.D. Thesis in Preparation, University of Waterloo, 2010.
21. **Yiu, R.** *Double-Blind Scores of an Object-Oriented Modelling Survey.* M.Math. thesis, University of Waterloo, 2000.