A Grand Challenge for Computing Research [GC-0]

Redressing the past: liberating computing as an experimental science

Meurig Beynon and Steve Russ
Department of Computer Science,
The University of Warwick,
Coventry CV4 7AL
{wmb,sbr}@dcs.warwick.ac.uk

In the conclusion to his paper 'Absolutely unsolvable problems and relatively undecidable propositions' [12] - submitted to a mathematics periodical in 1941 - Emil Post writes:

> ... perhaps the greatest service the present account could render would stem from its stressing of its final conclusion that mathematical thinking is, and must be, essentially creative. It is to the writer's continuing amazement that ten years after Gödel's remarkable achievement current views on the nature of mathematics are thereby affected only to the point of seeing the need of many formal systems, instead of a universal one. Rather has it seemed to us to be inevitable that these developments will result in a reversal of the entire axiomatic trend of the late nineteenth and early twentieth centuries, with a return to meaning and truth. Postulational thinking will then remain as but one phase of mathematical thinking.

Were Post alive today, he would find much in contemporary computing to endorse his perception of the future for formal systems. Theoretical computer science promotes a view of computing that has formal systems at its core, and its practice is dominated by the prolific invention of new formal systems designed to develop computer programs as mathematical entities. Computing in the wild by contrast is by and large an activity that defies formalisation, in which all manner of informal and heuristic techniques are used to construct programs with primary reference to their intended meaning and function. In the same way that Post's observation begs the question 'what then is the characteristic nature of mathematics, if not merely the study of formal systems?', so modern computing practice challenges us to identify the appropriate boundaries of computer science, and to recognise these as broader than the vision that the classical theory of computation alone can sustain.

In thinking about a science, it is tempting to identify it with its associated theory, rather than with the accepted practices that lead to the development of that theory. To do so is to neglect the very issue to which Post draws our attention: it is to give greater priority to the abstract study of formal systems within the discipline rather than the creative concrete activities that lead to their development. In fact, the maturity of a science has more to do with our understanding of the nature of its authentic - albeit informal - practices than with the quality of its theory. It is in this respect that computer science is a particularly immature discipline.

Computing is unusual in that it has evolved with an inheritance of theory that predates most of its practice. The pervasive use of computer technology has moreover served to reinforce established attitudes that privilege theory over practice. If we presume that all our experience is mediated by symbols and logic, can be cast in digital form, or accounted for by a suitably grand Theory of Everything, we cannot appreciate the full import of Post's conclusion: "... that mathematical thinking is, and must be, essentially creative". The inadequacy of a 'literary' view of science, that gives a more exalted status to the study of abstract symbolic representations than to interaction in the laboratory, is exposed by the philosopher of science David Gooding in his study of Faraday's researches in electromagnetism [6]. Gooding's analysis is helpful in identifying a perspective on computing that can embrace both its formal and informal aspects, and help in understanding its potential as an experimental science.

In broad terms, Gooding's concern is to show that Faraday's knowledge of electromagnetic phenomena, as it evolved through practical experiment and communication with other experimental scientists, was embodied in the physical artefacts and procedures for interaction, observation and interpretation that he developed, and that 'construals' [6] of this nature have an indispensable role in our appreciation of the science of electromagnetism. In this context, experiment has a significance far beyond that in our popular understanding of the scientific method (as expressed for instance in [15]: "One develops a theory that explains some aspect of reality, and then conducts experiments in order to provide evidence that the theory is right or demonstrate that it is wrong."). Though Faraday's experiments did eventually underpin Maxwell's mathematical theory, they initially had a far more primitive role. For instance, they served to distinguish transient effects from significant observables, and to relate Faraday's personal construals of a phenomenon to those of others who had typically employed different modes of observation and identified different concepts and terminology. Such experiments were not conducted post-theory to 'explain some aspect of reality', but rather to establish *pre-theory* what should be deemed to *be* an aspect of reality.

The question 'Is computing an experimental science?' has been topical in theoretical computer science circles ever since it was first posed by Milner in the inaugural lecture of the Laboratory for Foundations of Computer Science in 1986 [9]. Milner introduces his paper with reference to "a double thesis: that the design of computing systems can only properly succeed if it is well grounded in theory, and that the important concepts in a theory can only emerge through protracted exposure to application". Whilst Milner's thesis explicitly acknowledges the benefit that experimental practice can bring to pre-existing theory, it is not concerned with the role in computing – if any – for pre-theory experiment such as Faraday practised in physics. To address Post's agenda fully, it is necessary to identify computing as an experimental science in this more primitive sense.

Traditional sciences developed their roots in practice rather than in theory. Three features that are not fully represented in 'experiments to test theory' are characteristic of pre-theory experimentation in these fields. Each is concerned with a different aspect of gaining access to stable experience that can be coherently interpreted. Experience that is suitable for the articulation of a theory has to be:

- interpreted with respect to a preconceived context;

- circumscribed in respect of relevance;
- amenable to consistent interpretation.

In contrast, the construals with which an experimenter records and refines her current understanding through interaction pre-theory are:

- influenced by factors in the situation that are as yet unidentified;
- subject to interpretation and interaction in ways that are as yet unknown;
- capable of exposing inconsistencies and ambiguities that are as yet unresolved.

In the pre-theory context, these characteristics are qualities of construals as representational devices, relating to *situation*, *ignorance* and *nonsense* respectively, that are beyond the expressive scope of a formal system.

The above discussion helps to identify the following characteristics to be expected of computing as an experimental science:

- The philosophical orientation of such a science must admit semantic frameworks other than those based upon formal symbolic interpretation. It is evident that the way in which Faraday's construals 'represent' phenomena is quite different in character from that in which the symbols of a formal theory are given meaning. As Gooding emphasises, though a construal may make use of symbols whose meaning is encoded, it fulfils its representative function primarily because of the correspondence between the experimenter's manipulations of the construal and specific interactions with physical apparatus. The meaning of the construal cannot be appreciated in isolation from these interactions, which in many cases require the most skilful application of experimental techniques and may serve no more than a private and ephemeral role in helping to record and advance understanding. The somewhat polemical writings of the computer scientist Peter Naur are particularly relevant in this connection [10, 11], as they highlight the need to study interpretation and meaning with reference to what William James characterises as 'the personal stream of thought'[8].

- The science of computing must be founded upon principles that guide and characterise its practice. It may be appropriate to identify these principles with a 'scientific method', provided that its scope is broad enough to take account of situation, ignorance and nonsense. Brian Cantwell Smith, whose profound analysis of computing has spanned over twenty five years, concludes in [13] that 'computing is not an autonomous subject matter' – rather that 'the considerable and impressive body of practice associated with them amounts to … neither more nor less than the full-fledged social construction and development of intentional artefacts'. This conception is well-matched to Gooding's notions on the development of construals in physical science, but also accommodates the generalisation that is appropriate to computing, in which model-building is routinely concerned with representing phenomena that – unlike those in the natural world – are potentially artificial, imaginary and subjective in character.

- Whatever the science of computing is, it is most unlikely to be as yet unrepresented in practice. And - extrapolating from Post's observations about mathematics - it will be more closely associated with activities (such as validation) that are directly concerned with meaning and truth than with activities (such as verification) that primarily relate to formal systems. Candidate activities include the requirements modelling that precede formal specification, the development and use of spreadsheets in software development, and empirical processes such as user-interface design that involve experimentation and evaluation. The challenge to be faced is in relating informal practices to principles in a coherent fashion so as to enfranchise approaches to building computer programs without focusing upon their formal semantics.

Naturally enough, we are proposing the above conceptual framework for computing as an experimental science, and identifying its elaboration as a grand challenge, because we ourselves have already devoted much research to its realisation [1,2,3,4,5,16, 17]. The key idea in our research has been that of building computer-based construals to embody the patterns of agency and dependency amongst observables that we project on to the situations to which they refer. The correspondence between a situation and its construal is established and elaborated through interaction with the situation and with the construal, so that the interpretation of the construal is 'given in experience' rather than governed by a formal semantics. The principles by which such a construal is built to represent a situation are similar to - though more general than - those by which a spreadsheet is constructed and interpreted.

The three characteristics of computing as an experimental science identified above are all represented in our research. The principles, notations and tools that have been developed over the last twenty years provide the core of a final year undergraduate module and are illustrated by a web archive containing about 120 models [17]. Our research originated in practical model-building, and its philosophy, principles and tools continue to develop in response to innovative project work at undergraduate and postgraduate level. The primary emphasis has been on proof-of-concept and on making connections with other approaches to software development, especially those that exploit dependency maintenance. The most significant application of our approach to date has been that pioneered by Richard Cartwright at the BBC in connection with digital TV broadcasting. There are patent connections with the work of Gooding, and clear evidence of the generality that Cantwell Smith anticipates of 'intentional science'. In the spirit of Post's 'return to meaning and truth', our treatment of observables in model-building is reminiscent of mathematics prior to its formalisation in the 19th century [1, 14]. Like spreadsheets, our models can be viewed as capturing states as we experience them 'in the stream of thought', and their semantics is rooted in a fundamental tenet of James's radical empiricism – that primitive knowledge rests on the way in which 'one experience knows another' [5, 7].

More than sixty years after Post was writing, formal systems still exercise a major influence over academic views of computing. This prompts some to attach too much importance to the role of formal semantics, and others to associate principled approaches exclusively with inappropriate aspirations towards formalisation. It may well be that the culture of computing has played its part in promoting the modern infrastructure for managing university research, in which it is assumed that the quality of contributions to a subject can be assessed by quantitative metrics. This kind of development in respect of formal evaluation makes research of the kind proposed here vulnerable. Would Post have been amazed at this? Perhaps not. His paper – which included an extraordinary appendix in which he set out his own embryonic ideas about mathematics beyond formalisation – was rejected. One of the most important grand challenges for today's computing research community is surely to take note of Post's insight, and open its mind to the emergence of computing as an authentic experimental science.

References

1. W M Beynon and S B Russ, The Development and Use of Variables in Mathematics & Computer Science, in *The Mathematical Revolution Inspired by Computing*, IMA Conf Series 30, 285-95, 1991.
2. W M Beynon, J Rungrattanaubol, J Sinclair, Formal Specification from an Observation-Oriented Perspective, in J.UCS 6(4), 2000, http://www.jucs.org/jucs_6_4/formal_specification_from_an
3. W M Beynon, Liberating the Computer Arts, Proc DALI'2001, 1st Int. Conference of Digital and Academic Liberty of Information, University of Aizu, Japan, March 2001
4. W M Beynon, Empirical Modelling and the Foundations of Artificial Intelligence, Computation for Metaphors, Analogy and Agents, LNAI 1562, Springer, 322-364, 1999.
5. W M Beynon, Radical Empiricism, Empirical Modelling and the nature of knowing, in Proc WM2003 Workshop on Knowledge Management on Philosophy, Luzern, April 2003, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-85, ISSN 1613-0073
6. David Gooding, *Experiment and the Making of Meaning*, Kluwer Academic, 1990
7. William James, *Essays in Radical Empiricism*, Bison Books, 1996
8. William James, *Principles of Psychology*, Henry Holt USA, 1890, reprinted Dover, 1950
9. Robin Milner, Is computing an experimental science? Journal of Information Technology 2:58-66, 1987
10. Peter Naur, Knowing and the Mystique of Logic and Rules, Kluwer Academic, 1995
11. Peter Naur, The Anti-Philosophical Dictionary, naur.com, 2001
12. Emil Post, Absolutely unsolvable problems and relatively undecidable propositions, in M Davis, *The Undecidable*, Raven Press Books, 1965
13. Brian Cantwell Smith, *The Foundations of Computing*, http://www.ageofsig.org/people/bcsmith/print/smith-foundtns.pdf (13/7/02)
14. Steve Russ, The Mathematical Works of Bernard Bolzano, OUP 2004 (to appear)
15. Donald Sannella, What does the future hold for theoretical computer science? Proc. TAPSOFT97, Lille, France 1997
16. http://www.dcs.warwick.ac.uk/modelling
17. http://empublic.dcs.warwick.ac.uk/projects