2013-05

The Science in Computer Science

Denning, Peter J.

The Science in Computer Science (May 2013). Computer science is in a period of renaissance as it rediscovers its science roots.

http://hdl.handle.net/10945/35507

Prepublication Draft

DOI:10.1145/2447976.2447988

Peter J. Denning

Viewpoint
The Science in Computer Science

Computer science is in a period of renaissance as it rediscovers its science roots.

COMPUTER SCIENCE HAS for decades been ripped by an old saw: Any field that calls itself a science, cannot be science. The implied criticisms that we lack substance or hawk dubious results have been repeatedly refuted. And yet the criticism keeps coming up in contexts that matter to us.

It comes up in education in the debates about encouraging more student involvement in STEM (science, technology, engineering, and mathematics). Many critics see computer science mainly as technology or math. Will computer science be excluded because it is not seen as genuine science?

It comes up in research in debates about the predictive power of our analytic tools. In some subfields, such as storage management, performance prediction, and algorithms, experimental methods have led to reliable predictive models. In others, such as system safety and security, we lack predictive models and we can only speculate that experimental methods will lead to understanding. In his first ACM president's letter, Vint Cerf asks why software engineering does not rely more on experimental science (*Communications*, Oct. 2012). In so doing, he echoes a lament uncovered in a 1995 study of software engineering literature.[10] Do enough of us know the experimental methods needed to do this consistently well?

In interdisciplinary collaboration, it comes up when teams are formed and when credit is handed out. Why are computer scientists still often seen as professional coders rather than genuine collaborators?

My purpose here is to review the history of the question, "Is computing science?" and point to new answers that can help educators, researchers, and collaborators.

I use the term "computing" to refer to the set of related fields that deal with computation. These include computer science, computational science, information science, computer engineering, and software engineering. Interestingly, I have encountered less skepticism to the claim that "computing is science" than to "computer science is science."

A Short History of Science in Computing

Computing has been deeply involved in science since the beginning. A science vision pervaded the field through the 1950s, and then faded as technology development drew most of our energy through the 1980s. A science renaissance began in the 1990s, propelled by computational science and the discov-

CREDIT TK

ery of natural information processes. I will review each of these periods.

The pioneers who planned and built the first electronic computers were strongly motivated by visions of computers advancing science. The two most obvious ways were the numerical solution of mathematical models of physical processes, and the analysis of large datasets compiled from experiments. Computer science became a recognized academic field of study in 1962 with the founding of computer science departments at Purdue and Stanford. These departments maintained strong faculties in mathematical software, which directly supported science.

In 1967, Newell, Perlis, and Simon argued that the new field was a science concerned with all aspects of "phenomena surrounding computers."[12] However, many traditional scientists disagreed with the science claim; they held that true science deals with phenomena that occur in nature ("natural processes") whereas computers are man-made artifacts. Simon, a Nobel Laureate in economics, so strongly disagreed with the "natural interpretation" that he published a book *The Sciences of the Artificial* (MIT Press, 1969). He argued that economics and computer science met all the traditional criteria for science, and deserved to be called sciences even if, said Simon, their focal phenomena are "man-made as opposed to natural."

In the initial years of the field, most computing people devoted their energy to building the systems that could realize the visionary dreams of the founders. By the late 1970s, the computing industry was recruiting system people so vigorously that university departments were experiencing a "brain drain" of systems-oriented faculty. ACM leadership was very concerned: this trend threatened experimental computer science. I was deeply involved as ACM president in arguing the importance of experimental methods for computing and in assisting the U.S. National Science Foundation (NSF) to support experimental computer scientists. I wrote in 1980 that the experimental method (that is, science) is essential in computer science,[6] and in 1981 I cited the subfield of performance modeling and prediction as an exemplar of the ideals of science.[4] Despite these

efforts, many university departments lost their experimentalists and the science vision faded into the background.

In the 1980s, science visionaries from many fields saw ways to employ high-performance computers to solve "grand challenge" problems in science. They said computing is not only a tool for science, but also a *new method of thought and discovery in science*. (Aha! Computational thinking!) They defined computational science as a new branch of science imbued with this idea. The leaders of biology, epitomized by 1975 Nobel Laureate David Baltimore, went further, saying biology had become an information science and that DNA translation is a natural information process. Another biologist, Roseanne Sension, attributed the efficiency of photosynthesis to a quantum algorithm embedded in the cellular structure of plant leaves (*Nature*, April 2007). Biologists have thus been leaders in driving nails into the coffin of the "natural science" argument about computing. Many other scientists have reached similar conclusions. They include physicists working with quantum computation and quantum cryptography, chemists working with materials, cognitive scientists working with brain processes, economists working with economic systems, and social scientists working with networks.[9] All claimed to work with natural information processes. Stephen Wolfram went further, arguing that information processes underlie every natural process in the universe.[13]

Those two external factors—rise of computational science and discovery of natural information processes—have spawned a science renaissance in computing. Experimental methods have regained their stature because they are the only way to understand very complex systems and to discover the limits of heuristic problem solution methods.

Here is an example of an advance in algorithms obtained through an empirical approach. In May 2004, an international research group announced it had computed an optimal tour of 24,978 cities in Sweden (see http://tsp.gatech.edu/sweden). By iterating back and forth among several heuristic methods, they homed in on a provably optimal solution. Their computation took about one year on a bank of 96 parallel Intel Xeon 2.8GHz processors. With classical tour-enumeration algorithms, which are of order O(n!), the running time would be well beyond the remaining age of the universe. With experimental methods, algorithm scientists quickly found optimal or near-optimal solutions.

New fields heavily based in experimental methods have opened up—network science, social network science, design science, data mining, and Bayesian inference, to name a few. The widening claims that information processes occur in nature have refuted the notion that computer science is not "natural" and have complemented Simon's arguments that computing is a science of the artificial.

When is a Field a Science?

This brief history suggests that computing began as science, morphed into engineering for 30 years while it developed technology, and then entered a science renaissance about 20 years ago. Although computing had subfields that demonstrated the ideals of science, computing as a whole has only recently begun to embrace those ideals. Some new subfields such as network science, network social science, design science, and Web science, are still struggling to establish their credibility as sciences.

What are the criteria for credibility as science? A few years ago I compiled a list that included all the traditional ideals of science:[1,3]

- Organized to understand, exploit, and cope with a pervasive phenomenon.
- Encompasses natural and artificial processes of the phenomenon.
- Codified structured body of knowledge.
- Commitment to experimental methods for discovery and validation.
- Reproducibility of results.
- Falsifiability of hypotheses and models.
- Ability to make reliable predictions, some of which are surprising.

Although computing had subfields that demonstrated the ideals of science, computing as a whole has only recently begun to embrace those ideals.

Computing's original focal phenomenon was information processes generated by hardware and software. As computing discovered more and more natural information processes, the focus broadened to include "natural computation."[9] We can now say "computing is the study of information processes, artificial and natural."[1]

Computing is not alone in dealing with both natural and artificial processes. Biologists, for example, study artifacts including computational models of DNA translation, the design of organic memories, and genetically modified organisms (GMOs). All fields of science constantly face questions about whether knowledge gained from their artifacts carries over to their natural processes. Computing people face similar questions—for example, does studying a software model of a brain yield useful insights into brain processes? A great deal of careful experimental work is needed to answer such questions.

The question of "scienceness" of computing has always been complicated because of the strong presence of science, mathematics, and engineering in the roots and practice of the field.[8,11] The science perspective focuses on increasing understanding through experimental methods. The engineering perspective focuses on designing and constructing ever-improved computing systems. The mathematics perspective focuses on what can be deduced from accepted statements.

The term "theory" illustrates the different interpretations that arise in computing because of these three perspectives. In pure math, theory means the set of valid deductions from a set of axioms. In computing, theory more often means the use of formalism to advance understanding or design.

Effects on the Education System

Unfortunately, our education system for young people has not caught

up with these realities. From 2001 to 2009, college enrollments in CS majors dropped 50% (and are now recovering). From early analyses, we could see that students were losing interest in computing in high schools, half of which had no computer course at all, and many of the others relegated their one computer course to literacy in keyboarding and word processing. Very few had courses in the principles of computing. Around 1998, the U.S. Educational Testing Service wanted to help by focusing the Computer Science Advanced Placement (AP) curriculum on object-oriented programming. Unfortunately, the new AP curriculum did not help. Fewer than one-third of high schools actually used the CS AP curriculum and many teachers did not understand enough about object-oriented programming to teach it effectively.

Leaders in most of the STEM fields reported enrollment declines in the same period. Stimulating more student interest in STEM fields has become an international concern.

The science renaissance in computing has led to an explosion of new content on the principles of computing that is beginning to reach into high schools. With support from the U.S. National Science Foundation, a coalition of universities has defined a computer science principles introductory course and created prototypes (see http://csprinciples.org). The Educational Testing Service has embarked on a closely related project to redefine the AP curriculum around computing principles. Over the past two decades, Tim Bell of the University of Canterbury, New Zealand, has designed exercises and games for children 12–15 years old, allowing them to experience computing principles without using computers (see http://csunplugged.org). With my colleagues I have put together a presentation of all computer science principles (see http://great-principles.org).[2,5]

The dream articulated by Newell, Perlis, and Simon 50 years ago has come true. It endured many skeptical antagonists and weathered many storms along the way. Computing is now accepted as science. Some of us even believe computing is so pervasive that it qualifies as a new domain of science alongside the traditional domains of physical, life, and social sciences.[7] Educators are finding innovative ways to teach computing science to young people, who are now being infected with the magic, joy, and beauty of the field.

The question of whether computing is science is as old as the field.

Let Us Discuss

I am editor-in-chief of ACM's *Ubiquity*, an online peer-reviewed magazine about the future of computing and the people who are creating it. The *Ubiquity* editors put together a symposium of essays from 14 authors discussing various aspects of the question "Is computing science?" The authors include an ACM president, an ACM past president, two ACM A.M. Turing Award recipients, an NSF program manager, a journalist, six educators, and four interdisciplinary researchers. We drew five conclusions from the symposium.

First, the question of whether computing is science is as old as the field. It arose because traditional scientists did not recognize computational processes as natural processes. Even during the engineering years, when much of the energy of the field was devoted to building systems and understanding their theoretical limits, the field developed two important scientific theories. The theory of locality studied memory usage patterns of computations, and the theory of performance evaluation validated queueing network models for reliable performance predictions of computer systems.

Second, there is a growing consensus today that many of the issues we are studying are so complex that only an experimental approach will lead to understanding. The symposium documents advances in algorithmics, biology, social networking, software engineering, and cognitive science that use empirical methods to answer important questions.

Third, scientists in many fields now recognize the existence of natural information processes. This dismisses an early perception that CS deals solely with artificial information processes. Computing is not constrained to be a "science of the artificial." Computing is indeed a full science.

Fourth, because information processes are pervasive in all fields of science, computing is necessarily involved in all fields, and computational thinking has been accepted as a widely applicable problem-solving approach. Many students are now selecting computer science majors because it preserves their flexibility in choosing a career field later.

Fifth, computing presented as science is very engaging to middle and high school students. The science perspective expands well beyond the unfortunate and prevalent notion that computer science equals programming. A growing number of STEM teachers are embracing these new methods.

I invite you to look in at the full symposium and see for yourself what these people have said (see http://ubiquity.acm.org), and then weigh in with your own observations. C

References

1. Denning, P. Computing is a natural science. *Commun. ACM 50*, 7 (July 2007), 13–18.
2. Denning, P. Great principles of computing. *Commun. ACM 46*, 11 (Nov. 2003), 16–20.
3. Denning, P. Is computer science science? *Commun. ACM 48*, 4 (Apr. 2005), 27–31.
4. Denning, P. Performance analysis: Experimental computer science at its best. *Commun. ACM 24*, 11 (Nov. 1981), 725–727; http://doi.acm.org/10.1145/358790.358791.
5. Denning, P. The great principles of computing. *American Scientist 98* (Sept.–Oct. 2010), 369–372.
6. Denning, P. What is experimental computer science? *Commun. ACM 23*, 10 (Oct. 1980), 543–544; http://doi.acm.org/10.1145/359015.359016.
7. Denning, P. and Rosenbloom, P. Computing: The fourth great domain of science. *Commun. ACM 52*, 9 (Sept. 2009), 27–29.
8. Gonzalo. G. Is computer science truly scientific? *Commun. ACM 53*, 7 (July 2010), 37-39. http://doi.acm.org/10.1145/1785414.1785431.
9. Kari, L. and Rozenberg, G. The many facets of natural computing. *Commun. ACM 51*, 10 (Oct. 2008), 72–83; http://doi.acm.org/10.1145/1400181.1400200.
10. Lukowicz, P., Tichy, W., Pechelt, L., and Heinz, E. Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software 28*, 1 (Jan. 1995), 9–18.
11. Morrison, C. and Snodgrass, R.T. Computer science can use more science. *Commun. ACM 54*, 6 (June 2011), 36–38; http://doi.acm.org/10.1145/1953122.1953139.
12. Newell, A., Perlis, A., and Simon, H. Computer science. *Science 157* (1967), 1373–1374.
13. Wolfram, S. *A New Kind of Science*. Wolfram Media, 2002.

Peter J. Denning (pjd@nps.edu) is Distinguished Professor of Computer Science and Director of the Cebrowski Institute for information innovation at the Naval Postgraduate School in Monterey, CA, is Editor of ACM *Ubiquity*, and is a past president of ACM.