Peter J. Denning

Computing is a Natural Science

Information processes and computation continue to be found abundantly in the deep structures of many fields. Computing is not—in fact, never was—a science only of the artificial.

Computing is now a natural science. Computation and information processes have been discovered in the deep structures of many fields. Computation was present long before computers were invented, but the remarkable shift to this realization occurred only in the last decade. We have lived for so long in the belief that computing is a science of the artificial, it may be difficult to accept that many scientists now see information processes abundantly in nature.

Revolution in the Making

This revolution has been gestating for a long time. Its three main stages were tools (beginning in the 1940s), methods (beginning in the 1980s), and fundamental processes (beginning in the 2000s).

In the 1940s, the era of the first electronic digital computers, computation was seen as a tool for solving equations, cracking codes, analyzing data, managing business processes, running simulations, and solving models. Computation soon established itself as a powerful tool that made formerly intractable analyses tractable. It took many technologies to new heights, such as atomic energy, advanced aircraft and ship design, drug design, structural analyses of buildings, and weather prediction.

By the 1980s, computation had become utterly indispensable in many fields. It had advanced from a tool to exploit existing knowledge to a means of discovering new knowledge. Nobel Physics Laureate Ken Wilson was among the first to say that computation had become a third leg of science, joining the traditions of theory and experiment. He and others coined the term "computational science" to refer to the search for new discoveries using computation as the main method. This idea was so powerful that, in 1989, the U.S. Congress passed into law the High Performance Computing and Communication Initiative to stimulate technological advances through high-performance computation.

By 2000, computation had advanced further. Scientists from many fields were saying they had discovered information processes in the deep structures of their

SERGE BLOCH

fields. Nobel Laureate and Caltech President David Baltimore commented: "Biology is today an information science. The output of the system, the mechanics of life, are encoded in a digital medium and read out by a series of reading heads. Biology is no longer solely the province of the small laboratory. Contributions come from many directions." (*The Invisible Future*, Wiley, 2001, p. 45.)

Baltimore was saying that nature long ago learned how to encode information about organisms in DNA and then to generate new organisms from DNA through its own computational methods. Biologists and computer scientists today collaborate closely as they seek to understand, and eventually to influence, those natural information processes.

Biology was not the only field to say this. Physicists said that quantum waves carry information that generates physical effects. They have made significant advances with quantum computation and quantum cryptography. Nobel Laureate Richard Feynman became famous for showing that quantum electrodynamics (QED) was nature's computational method for combining quantum particle interactions. In the early 1980s, computational scientists at NASA-Ames discovered a successful, methane-resistant heat shield material for the Jupiter Probe by computing its molecular structure from the Schroedinger Equation. In his book *A New Kind of Science* (2002), Stephen Wolfram proclaimed that nature is written in the language of computation, challenging Galileo's claim that it is written in mathematics.

Economists analyze economic systems for their inherent information flows. Management scientists claim workflow, commitments, and social networks as fundamental information processes in all organizations. Artists and humanists use computation for everything from analysis to the creation of new works. Web researchers have discovered new social behaviors and ways of computing by using the entire Web as their laboratory. Computing artifacts have become matters of style and culture (iPod, eBay, Wikipedia, Google, Playstation, Xbox, Wii, and much more). Even politicians are utilizing sophisticated social data analyses, computational gerrymandering, and blogging. Jeanette Wing has concluded that computational concepts are deeply embedded into everyday thinking in many fields [10]. Computation is everywhere.

Although the acceptance of computation in many fields is new, the acceptance of information is not. Information has been a key concept in many fields since 1948 [7]. Norbert Weiner said in 1958, "Cybernetics is the science of communication and control, whether in machines or living organisms." Cybernetics did not survive as a science because few people were willing to accept Weiner's claim that his new science was somehow more encompassing than theirs.

> The old definition of computer science—the study of phenomena surrounding computers—is now obsolete. Computing is the study of natural and artificial information processes.

This acceptance of computing as science is a recent development. In 1983, Richard Feynman told his Caltech students: "Computer science differs from physics in that it is not actually a science. It does not study natural objects. Neither is it mathematics. It's like engineering—about getting to do something, rather than dealing with abstractions." (*Lectures on Computation*, Addison-Wesley, 1996, p. xiii.)

Feynman's idea was consistent with the computational science view at the time. Less than a generation later, his colleagues had come to see information processes as natural occurrences and computers as tools to help study them.

This is a striking shift. For a long period of time many physicists and scientists claimed that information processes are manmade phenomena of manmade computers. The old definition of computer science—the study of phenomena surrounding computers—is now obsolete. Computing is the study of natural and artificial information processes. Computing includes computer science, computer engineering, software engineering, information technology, information science, and information systems.

Principles Framework

In the mid-1990s, it seemed that the computing field had matured to the point where it was possible to articulate its fundamental principles, and I began experimenting with frameworks that do this. In 2003, in this column I launched a campaign to develop a principles framework for computing [3, 4]. The significant benefits of accomplishing this include:

- Revealing the deep structure of computation and why it permeates so many other fields;
- Revealing common principles among technologies, enabling simplification, new discoveries, and innovations;
- Giving a common language for discussing computation with other fields;
- Inspiring new approaches to teaching and learning computing; and
- Inspiring young people.

The fundamental questions addressed by a principles framework are:

- What is information?
- What is computation?
- How does computation expand what we know?
- How does computation limit what we can know?

Like biology's question, "What is life?", these questions are asked in every new situation. The current version of the framework is available for inspection and comments at the Great Principles (GP) Web site [6].

Articulating a framework turned out to be much more difficult than any of us thought it would be. The reason was that we have had no serious community discussion of our fundamental principles. We literally did not know how to articulate some of our deepest principles. Our initial attempts to formulate a principles framework produced little more than rearrangements of the technology lists in the ACM curriculum body of knowledge. But eventually, we arrived at something new: a top-level framework of seven (overlapping) categories of principles that cut across many technologies:

- Computation (meaning and limits of computation);
- Communication (reliable data transmission);
- Coordination (cooperation among networked entities);
- Recollection (storage and retrieval of information);
- Automation (meaning and limits of automation);
- Evaluation (performance prediction and capacity planning); and
- Design (building reliable software systems)

These categories cover the main functions of computing systems. While the numbers of new technologies and new principles are on the rise, the number of categories is likely to remain stable for a long time.

These categories are windows into a single computing knowledge space rather than slices of the space into separate pieces. Each window sees the space in a distinctive way; the same thing can be seen in more than one window. Internet protocols, for example, are sometimes seen as means for data communication, sometimes as means of coordination, and sometimes as means for recollection of data.

We found that most computing technologies draw principles from all seven categories. This finding confirms our suspicion that a principles interpretation will help us see many common factors among technologies.

Computing interacts constantly with other fields. The other fields teach us more about computing, and we help them find better ways to understand the world. The interplay is difficult to accommodate in our traditional definitions, which tie computation to the execution of algorithms on a computer. It is not difficult in the GP framework, which says that a computation is a sequence of representations, in which each transition is controlled by a representation. By this definition, DNA can compute. The computer is the tool, computation is the principle.

The table here is a sampler with a principle from each category, along with examples from within computing and from the rest of the world.

Principle	Summary	Computing Examples	
Intractability (Computation)	Over 3,000 key problems in science, engineering, and commerce require more computation, even for small inputs, than can be done in the life of the universe.	Searching for optimal solutions. Traveling salesman. Knapsack packing. Bin packing. Tiling a plane.	Parcel delivery. Truck transportation. Taxi routing. Airline routing. Scheduling (industrial engineering).
Compression (Communication)	Representations of data and algorithms can be significantly compressed and the most valuable information recovered later.	Compression of voice (MP3, MP4, ACC), images (JPEG, GIF), files (Zip). Fourier transform.	Operation of cochlea in the ear. Morse code.
Choosing (Coordination)	An uncertainty principle: it is not possible to make an unambiguous choice of one of several alternatives within a fixed deadline.	Hardware that never crashes while responding to interrupts. Mutual exclusion. Deadlocks.	Traffic control. Telephone and network routers. DNA sequencing. Free will (psychology).
Locality (Recollection)	Computations cluster their information recall actions into hierarchically aggregated regions of space and time for extended periods.	Virtual memory. Hardware caching. Web caching. Interconnection structures in parallel machines.	Functional brain cell clusters. Near decomposable economic systems. Punctuated equilibrium (biology).
Search (Automation)	Finding a pattern or configuration in a very large space of possibilities.	Genetic algorithms. Evolutionary computing. Branch and bound. Gradient search.	Genetic evolution. Passing of genes to descendents.
Bottlenecks (Evaluation)	Forced flow laws: in any network, the throughput at any node is the product of the network throughput and the visits per task to the node.	Saturation and bottlenecks in communication networks.	Fast propagating urban gridlock. Assembly lines (industrial engineering).
Hierarchical Aggregation (Design)	Larger entities are composed of many smaller ones.	OS and network software levels. Information hiding. Modularity. Abstraction.	Ladder of scale (astronomy and physics). Functional organs (biology). Fractals.

Examples of principles (from [6]).

Future Directions of Computing

Computing is evolving constantly. New principles are discovered; older principles fall out of use. An example of a new principle is the scale-free structure of network connectivity; an example of an out-of-use principle is the guideline for vacuum tube logic circuits. To help monitor the evolution of the field and find new principles-based connections among technologies and fields, the GP Web site contemplates a Great Principles Library, an evolving collection of materials, tools, and editorial process to support the learning, teaching, application, and cross linking of technologies and principles [6].

There is a trend in the computing field involving games. Not only is the video game industry pursuing it, but business and military organizations are turning to virtual reality simulation games as effective training grounds for various skills (as indicated in this month's special section). Dozens of universities have established BS or MS degrees in gaming. Is this a deep trend? Or just a fad?

The framework helps us answer. In the category of coordination, a game is a model for rules of interactions governing complex adaptive social-technical systems. As far as we can tell, this interpretation of game is the most general we have to describe all instances of coordination [6]. In his book, James Carse explores the amazing depth of the game interpretation, beginning with this tantalizing statement: "There are at least two kinds of games. One could be called finite, the other, infinite. A finite game is played for the purpose of winning, an infinite game for the purpose of continuing the play." (*Finite and Infinite Games*, Ballantine, 1986, p. 1.)

Carse's finite game bears a striking resembling to our notion of closed (terminating) computation, and infinite game to open (nonterminating) computation. Not only are we moving away from closed to open computations as objects of study, we are engaging new fields as infinite rather than finite games. Examples:

- Theoretical computer science is moving away from closed com-

> The notion that there are principles that transcend computers and apply to computation in all fields is already moving into education, where it is producing innovative ways to teach computing and is inspiring young people to consider computing majors.

putation and toward interactive computation [5].

- Considerable information is accessible to the Web through database interfaces that cannot be queried by search engines. Some estimates put the amount of searchable data at less than 1% of the accessible Web. Social and political science researchers are studying the Web space as a game in which new policies might alter the play to make more of the accessible data searchable.
- Evolving knowledge communities such as eBay, Web, Google, iTunes, Wikipedia, Blogosphere, Amazon.com, Amazon Turk, and crowdsourcing have become the research laboratories for innovations, social networking, trust, influence, and power.
- The Web and Internet, both infinite games, are opening up new areas of science on account of computation. A group of researchers has recently named this area "Web science" [2, 8]. In just one example, the statistical mechanics of scale-free networks accounts for structures humans generate in the Web and the success of many strategies for redundancy, search, social networking, and knowledge discovery.
- Luis von Ahn of Carnegie Mellon University has defined a category of games called "human computations." As a by-product of the play, the game produces useful results for which there is no known algorithm. The first example of the genre is espgame.com, which labels images with accurate keywords. It presents an image to random pairs of players, who must agree on a word that describes the image without seeing what the other is proposing. The output of the game is a growing database of accurately labeled images that has already greatly improved Google's image searches.

A similar shift is occurring in the other sciences. Our examples from biology, physics, materials science, economics, and management science show that they have moved beyond computing as a description of their information processes to a malleable generator of ongoing new behaviors.

Teaching and Learning

The notion that there are principles that transcend computers and apply to computation in all fields is already moving into education, where it is producing innovative ways to teach computing and is inspiring young people to consider computing majors.

An early U.S. example was the 1999 National Research Council report, *Being Fluent in Information Technology*. The objective was to define "what everyone should know about information technology." Larry Snyder of the University of Washington, who chaired the study group, wrote a widely used textbook that helps almost anyone learn to be fluent in computing [9].

A team led by Tim Bell at the University of Canterbury in New Zealand developed *Computer Science Unplugged* [1], a way to understand computing concepts without a computer. With games, exercises, and magic tricks they teach children computing principles using ordinary materials such as cards, drawing paper, and whiteboards. For example, they teach binary numbers by having children build numbers from cards with 1, 2, 4, and 8 dots on them. Their approach inspires curiosity and excitement among children. The subtle genius of their approach is exposing how many computing concepts don't need computers.

The Canterbury team recently

joined with UCLA, the University of Washington, and Carnegie Mellon University in a consortium (CS4ALL) to propagate the ideas to a much larger audience of students and teachers. They organized summer workshops for students to take them through the "unplugged" material and develop new material.

The Society for Amateur Scientists, led by Shawn Carlson, has developed an extensive program to help children learn the basic principles, values, and practices of science. They help children with science fair projects (scifair.org) and participation in LabRats, a scouts-like science community (labrats.org).

The GP framework complements these efforts by giving a complete map of computing principles and a language to discuss them with other fields.

Conclusion

The long-awaited computation revolution now envelops us. Information and computation are being discovered as fundamental processes in many fields. Computing is no longer a science of just the artificial. It is the study of information processes, natural and artificial.

The great principles framework supports our continuing play in the game of advancing computing and linking it with other fields. The more we learn, the less distance we see between us and other fields.

The rise of interest in games in computing is no accident, especially when games are seen as models for large complex adaptive systems that never terminate. Computing is an infinite game.

The revolution can give heart to those concerned about the current enrollment crisis, and to those worried that computer science is dying. The current crisis will strengthen us because it will stimulate much curriculum innovation and is likely to draw many bright people into the field. **C**

References

1. Bell, T., Whitten, I., and Powell, M. Computer science unplugged; www.unplugged.canterbury.ac.nz.
2. Berners-Lee, T., Hall, W., Hendler, J., Shadbolt, N., and Weitzner, D. Creating a science of the Web. *Science 313* (Aug. 11, 2006), 769–771.
3. Denning, P. Great principles of computing. *Commun. ACM 46*, 11 (Nov. 2003), 15–20.
4. Denning, P. Is computer science science? *Commun. ACM 48*, 4 (Apr. 2005), 27–31.
5. Goldin, D., Smolka, S., and Wegner, P. *Interactive Computation: The New Paradigm*. Springer, 2006.
6. Great Principles of Computing Web site; http://cs.gmu.edu/cne/pjd/GP.
7. Kline, R.R. Cybernetics, management science, and technology policy: The emergence of "information technology" as a keyword, 1948–1985. *Technology and Culture 47* (June 2006), 513–535; http://muse.jhu.edu/journals/technology_and_culture/v047/47.3kline.html
8. Microsoft. Towards 2020 science (2006); http://research.microsoft.com/towards2020science.
9. Snyder, L. *Fluency with Information Technology*. Addison-Wesley (second edition, 2005).
10. Wing, J. Computational thinking. *Commun. ACM 49*, 3 (Mar. 2006), 33–35.

Peter J. Denning (pjd@nps.edu) is the director of the Cebrowski Institute for Innovation and Information Superiority at the Naval Postgraduate School in Monterey, CA, and a past president of ACM.