The profession of IT – Computing: the fourth great domain of science.

2 authors, including:

Paul Rosenbloom
University of Southern California
188 PUBLICATIONS **8,717** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   The Sigma Cognitive Architecture and System View project

Project   Computing as a Great Scientific Domain View project

The Profession of IT
Computing: The Fourth Great Domain of Science

Computing is as fundamental as the physical, life, and social sciences.

THERE IS MUCH discussion about the generality and pervasiveness of computing. Is computing really an inescapable part of the world? What does that imply about science? Engineering? Education? How can we build new stories and education experiences that attract new young people to the field? Can computing, like other sciences, advance technology and applications through strong scientific advances? Can computer science rightfully claim a place at the table of science? And so on.

Many people have been warming up to the ideas that computing is science, deserves a place at the table of science, and is a rewarding profession. Yet a question nags at the edge of perception. Computers are admittedly everywhere. Roads, electricity, radio, television, and food are everywhere too, but they are not science. They are infrastructure. Why is computing any different?

We recently discovered a new answer to this old question. We noticed that all the acknowledged sciences are grouped into three great domains: physical, life, and social. We asked, what makes them great domains of science? And we found that computing meets all the same criteria. In other words, computing is the fourth great domain of science.
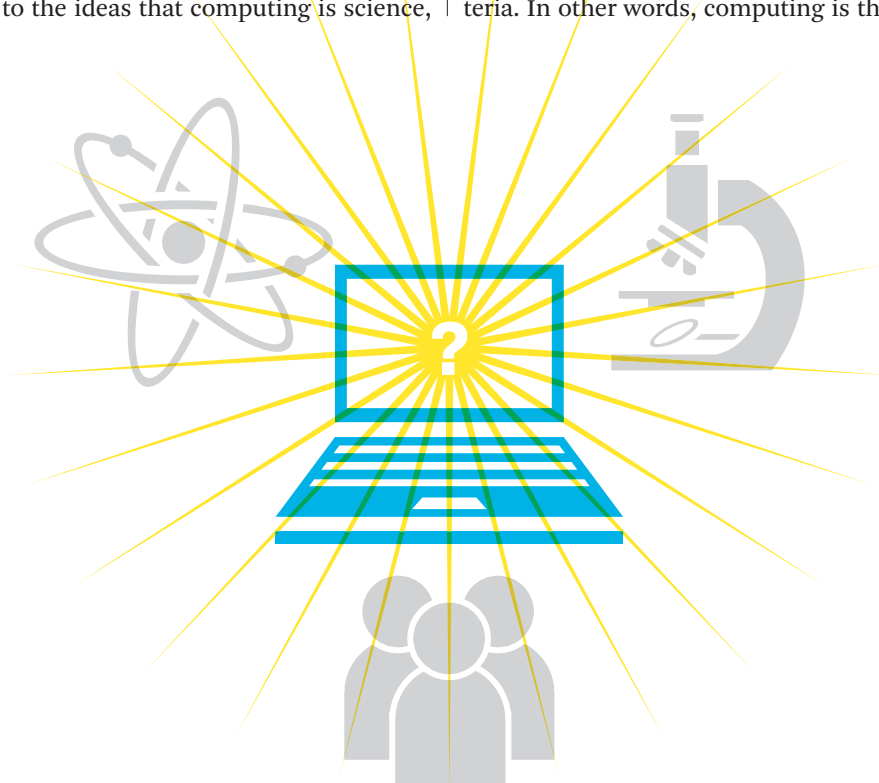
We will show you why we make this claim. We hope that you will not only want to discuss it, but that you will warm up to it too.

Great Scientific Domains

Most of us understand science as the quest to understand what is so about the world. Through observation and experimentation, scientists seek to discover recurrent phenomena. They formulate models to capture recurrences and enable predictions, and they seek to validate or refute models through experiments. Much of computing conforms to these ideals.[7,10]

Science has a long-standing tradition of grouping fields into three categories: the physical, life, and social sciences. The physical sciences focus on physical phenomena, especially materials, energy, electromagnetism, gravity, motion, and quantum effects. The life sciences focus on living things, especially species, metabolism, reproduction, and evolution. The social sciences focus on human behavior, mind, economic, and social interactions.[8] We use the term "great domains of science" for these categories.[9]

These domains share three common features: their foci are distinctive phenomena important in all sciences; the fields of each category have rich sets of structures and processes that evolve together through constant interaction; and their influence is extensive, touching all parts of life and providing unique and useful perspectives.

Examples of computing interacting with other domains.[8]

	Physical	Social	Life	Computing
Implemented by	mechanical, optical, electronic, quantum, and chemical computing	Wizard of Oz, mechanical turks, human cognition	genomic, neural, immunological, DNA transcription, evolutionary computing	compilers, OS, emulation, reflection, abstractions, procedures, architectures, languages
Implements	modeling, simulation, databases, data systems, digital physics, quantum cryptography	artificial intelligence, cognitive modeling, virtual humans, autonomous systems	artificial life, biomimetics, systems biology	
Influenced by	sensors, scanners, computer vision, optical character recognition, localization	mouse, keyboard, learning, programming, user modeling, authorization, speech understanding	eye, gesture, expression, and movement tracking, biosensors	networking, security, parallel computing, distributed systems, grids
Influences	locomotion, fabrication, manipulation, open-loop control	screens, printers, graphics, speech generation, network science, cognitive augmentation	bioeffectors, haptics, sensory immersion	
Bidirectional Influence	robots, closed-loop control	human-computer interaction, full immersion, games	brain-computer interfaces	

Computing does not seem to fit nicely into any of the traditional domains. Computation is realized in physical media and is even part of some physical processes (for example, quantum mechanical waves). More recently computation has been found in living systems (for example, DNA transcription) and social systems (for example, evolution of scale-free networks). Although computational methods are used extensively in all the domains, none studies computation per se—computation is not a physical effect, a living entity, or a social entity.

What if computing is a separate domain? It satisfies the three criteria. It

> **Computing interacts not only with people and other living systems, but with the physical world.**

has a distinctive focus—computation and information processes. Its constituent fields—computer science, informatics, information technology, computer engineering, software engineering, and information systems—and its structures and processes are in constant interaction. Its influence is pervasive, reaching deep into people's lives and work.

The core phenomena of the computing sciences domain—computation, communication, coordination, recollection, automation, evaluation, and design[2,6]—apply universally, whether in the artificial information processes generated by computers or in the natural information processes found in the other domains. Thus, information processes in quantum physics, materials science, chemistry, biology, genetics, business, organizations, economics, psychology, and mind are all subject to the same space and time limitations predicted by universal Turing machines. That fact underpins many of the interactions between computing and the other fields and underlies the recent claim that computing is a science of both the natural and the artificial.[3-5]

It might be asked whether mathematics is a great domain of science. Although mathematics is clearly a great domain, it has traditionally not been considered a science.

The Nature of Interactions

Two out of the three criteria listed earlier involve interactions, either among structures and processes or among domains. These interactions generate the essential richness of science. They also complicate how we observe and understand science.

Hierarchical taxonomies are the usual ways of observing a domain of science. It is easy to craft a tree hierarchy representing all the parent and child relationships among fields in the domain. For example, the physical sciences are partitioned into chemistry, physics, astronomy, geology, etc., and each of those may be partitioned further, for example, regular and organic chemistry. Each field has its own "body of knowledge," often represented with a taxonomy or a tree. Computing is likewise divided into constituent fields and subfields, each with a body of knowledge.

Hierarchical structures are very good for understanding static aspects of a field, but not its dynamics. Within a field, the interesting phenomena are not simply the properties of "things"; they are interactions among multiple things. Chemistry is not simply chemicals; it is the reactions among elements. Mechanics is not simply gears and levers; it is the forces among these parts. Psychology is not simply emotions, urges, and mental states; it is transactions and relationships. Similarly, computing is not just algorithms and data structures; it is transformations of representations.

The interactions are the real action of a field. Their complexities and uncertainties demand constant experimentation and validation in science and engineering. They make things messy and unpredictable. They are sources of innovation.

Scientific phenomena can affect each other in one of two ways: *implementation* and *interaction*. A combination of existing things implements a phenomenon by generating the intended behaviors of the phenomenon. Digital hardware physically implements computation.

Artificial intelligence implements aspects of human thought. A compiler implements a high-level language with machine code. In chemistry, hydrogen and oxygen implement water. In molecular biology, complex combinations of amino acids implement life.

Interaction occurs when two phenomena influence each other. In physics, atoms arise from interactions among the forces generated by protons, neutrons, and electrons. In astronomy, galaxies interact via gravitational waves. In computing, humans interact with computers.

Interactions exist not only within domains but across domains. Computing is implemented not only by physical processes, but by life processes (for example, DNA computing) and social processes (for example, games that produce outputs[1]). Likewise, computing can implement, or at least simulate, structures and processes in these other domains. Computing interacts not only with people and other living systems, but with the physical world (for example, through sensor networks and robots).

The accompanying table illustrates a wide range of implementation and interaction relationships between computing and the four domains (including itself). An entry in the table identifies a way that computing interacts with a domain. For example, the entry for "quantum computing" in the "Physical" column and "implemented by" row means that computation is implemented by quantum processes from the physical sciences domain.

The examples in the table are sufficient to demonstrate the amazing extent of interactions between computing and the other domains. Computing is much more than infrastructure; it is an equal partner that strongly influences thought, practice, and approach.

There is still more to the story. Many other interactions involve more than two fields or domains. For example, the emerging field of "network science" is built on multi-way interactions among the computing, physical, and social sciences.

Down to the Basics

We've used the term "computation" as if everyone agrees on its meaning. In fact, this is not so. Typical definitions include "activity of a computer," "phe-

> **To say that computing is a domain of science does not conflict with computing's status as a field of engineering or even mathematics.**

nomena surrounding computers," and "transformation of content." None of these captures all the notions of computing we can see at work in the table. For example, biologists believe DNA encodes information about the living body and that DNA transcription is a natural information-transforming process that creates the amino acids that generate new life. They clearly do not think of information as something stored in a computer database or transcription as an activity of a computer.

One way to define computing in a sense broad enough to cover everything in the table is to base it on the evolution of representations.[6] Except in artificial intelligence, representations are a somewhat neglected aspect of computing. Computing scientists need to get better answers to key questions. What do we mean by a representation? What does it mean to represent something in a computationally amenable format? Should representations be grounded in the world, or projected from a mathematical definition? What is not a representation? In what way is computation an evolution of representations?

In fact, representations are not the only fundamental principle of computing in need of new answers. Many of the oldest questions are being reopened.[11] What is computation? What is information? What is intelligence? How can we build complex systems simply?

Conclusion

Computing is pervasive because it is a fundamental way of approaching the world that helps understand its own cru-

cial questions while also assisting other domains advance their understandings of the world. Understanding computing as a great domain of science will help to achieve better explanations of computing, increase the attraction of the field to newcomers, and demonstrate parity with other fields of science.

To say that computing is a domain of science does not conflict with computing's status as a field of engineering or even mathematics. Computing has large slices that qualify as science, engineering, and mathematics. No one of those slices tells the whole story of the field.

The exercise of examining computing as a domain of science reveals that the extent of computing's reach and influence cannot be seen without a map that explicitly displays the modes of implementation and interaction. It also reveals that we need to revisit deep questions in computing because our standard answers, developed for computer scientists, do not apply to other fields of science. Finally, it confirms that computing principles are distinct from the principles of the other domains. ▣

References
1. von Ahn, L. Games with a purpose. *IEEE Computer Magazine* (June 2006), 96–98.
2. Denning, P. Great principles of computing. *Commun. ACM 46*, 11 (Nov. 2003), 15–20.
3. Denning, P. Is computer science science? *Commun. ACM 48*, 4 (Apr. 2005), 27–31.
4. Denning, P. Computing is a natural science. *Commun. ACM 50*, 7 (July 2007), 13–18.
5. Denning, P. Beyond computational thinking. *Commun. ACM 52*, 6 (June 2009), 28–30.
6. Denning, P. and Martell, C. Great Principles of Computing Project; http://cs.gmu.edu/cne/pjd/GP
7. Newell, A., Perlis, A.J., and Simon, H.A. "Computer Science," letter in *Science 157*, 3795 (Sept. 1967), 1373–1374.
8. Rosenbloom, P.S. A new framework for computer science and engineering. *IEEE Computer* (Nov. 2004), 31–36.
9. Rosenbloom, P.S. *Computing as a Great Scientific Domain: A Multidisciplinary Perspective.* To be published (2009).
10. Simon, H.A. *Sciences of the Artificial, Third Edition.* MIT Press, Boston, MA, 1996.
11. Wing, J. Five deep questions in computing. *Commun. ACM 51*, 1 (Jan. 2008), 58–60.

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

Paul S. Rosenbloom (rosenbloom@usc.edu) is a professor of computer science in the Viterbi School of Engineering at the University of Southern California, a project leader at the USC institute for Creative Technologies, and the former deputy director of the USC Information Sciences Institute.