Leveraging Computing Sciences in STEM Education

Paul S. Rosenbloom
Department of Computer Science and Institute for Creative Technologies
University of Southern California
12015 Waterfront Dr.
Playa Vista, CA 90094
rosenbloom@usc.edu

In this short essay, I make three claims:

1. It is time to put the final nails in the coffin of the *argument from artificiality*, that computing isn't a true science because it studies artificial rather than natural phenomena.
2. It is time to go beyond the straightforward conclusion that computer science is a respectable scientific discipline – such as physics or chemistry – to the bolder conclusion that computing actually constitutes an entire domain of science. Let us call this domain the *computing sciences*. The computing sciences are the equal of the physical, life and social sciences.
3. The domain of the computing sciences opens up new worlds of subject matter for STEM education.

These claims are grounded in a decade of explorations into the nature, structure, stature and role of computing in the sciences. The results of these explorations are recorded in several papers – including (Rosenbloom, 2004) and (Denning & Rosenbloom, 2009) – and in the new book *On Computing: The Fourth Great Scientific Domain* (Rosenbloom, 2012).

In the opening statement to this symposium, Denning introduces the argument from artificiality, and discusses two main counterarguments: science is possible in artificial domains; and natural forms of computing have been identified. But two additional counterarguments are also worth mentioning. First, the distinction between natural and artificial is itself artificial and increasingly meaningless. Why are structures that are created by physical processes (such as planets and rivers) and various species of animal (such as anthills and beaver dams) considered natural, while comparable structures created by people are considered artificial? This appears to stem from the traditional notion that humans occupy a special status outside of nature, with their products therefore also being outside of nature. Yet we now understand that people are as much a part of nature as any other biological organisms, and that their products – including human-created forms of computation – are likewise just as much a part of nature. The distinction between natural and artificial phenomena, while still useful in some contexts, is thus no longer defensible as a means of distinguishing science from non-science.

The second additional counterargument is that our continually improving abilities to modify natural processes at finer levels of detail makes it increasingly

To appear in *ACM Ubiquity*.

difficult across the board to distinguish what is due to human agency and what is not. The only difference between natural and artificial flavorings is the feedstock; the molecules themselves are identical. Genetically modified plants are changed by human agency but are still plants. A liver grown from stem cells under laboratory-controlled conditions is a full biological organ. As with computing, much of the rest of science needs increasingly to be concerned with both natural and human-created (or human-modified) forms. Computing is thus becoming indistinguishable from the other sciences in terms of the nature of what it studies; and science as a whole is on a path to seek understanding of everything around and in us regardless of its genesis.

Each of the existing domains of science studies the interactions among a distinctive set of structures and processes. The physical sciences study physical (non-biological) structures – from atoms and molecules to rocks, planets and galaxies – plus the processes that operate on them. The life sciences study biological organisms and the processes of metabolism, growth, development, reproduction and aging that operate on them. The social sciences study the thought processes and behaviors of individuals and groups of people. As discussed in an earlier *Ubiquity* symposium (2010), the computing sciences study *information and its transformation*. The field took an important step beginning in the 1990s as it realized that its proper subject for study was information and its transformation rather than simply (human-made) computers. The former emphasis on computers probably blinded us to the existence of natural forms of computing. Now we can see that various forms of computing are already embodied, for example, within living organisms.

Information can support representations and make distinctions. It is at the heart of the knowledge in our heads and of everything humans have written down on paper. It exists in many natural structures, as well as in the memories and data structures used by computers. Information is also at the core of communications, whether in modern computer networks or earlier oral, written or wired traditions. The study of information has its roots in philosophy and mathematics, and it has always played an important role in the humanities. It has more recently led to the creation of such disciplines as information theory and information science. Many parts of computer science – from data structures and databases to web technologies and artificial intelligence – also play a key role in the understanding of information. What distinguishes computer science from all of these other approaches is its emphasis on the *transformation* of information. This is why the experimental method is so important within computer science. Structures by themselves, whether informational or otherwise, often lend themselves to analytical forms of understanding. But when complex processes interact with structures, analytical methods typically fall short and experimental methods are the only means toward understanding.

If computing is science, where does it fit in the traditional taxonomy of the physical, life, and social sciences? The answer is that it does not fit. None of the physical, life, and social science domains has claimed computing sciences simply because none of them is concerned with information and its transformation. Thus computing seems to be an orphan. My argument, laid out in *On Computing*, is that computing is not an orphan but qualifies as a domain of science in its own right that is on a par with the other three.

On Computing also introduces in some detail the *relational approach* to understanding the relationships between the core subject matter of computing and the cores of the other domains. Much of the book then analyzes how a pair of relationships among domains – *implementation* and *interaction* – can illuminate the scope and organization of computing as a scientific domain, while also highlighting the rich space of exciting multidisciplinary topics that are too often relegated to the fringes of the field. The analysis flows through four stages:

1. Monadic computing – computing in isolation – as in theoretical computer science
2. Pure dyadic computing – computing relating to itself – as when compilers implement one language in another and when multiple computers interact in networks
3. Mixed dyadic computing – computing relating to individual other domains – as in robotics and computational science
4. Polyadic computing – computing relating to multiple other domains – as in mixed reality and ubiquitous computing

It is perhaps the multidisciplinary topics that have the most to offer STEM education, as they shift computing out of the abstract space of pure information, to make contact with aspects of the world familiar to students. The interdisciplinary topics are the home of the current and future action in computing. Students find these topics exciting. Moreover, students who have worked with multidisciplinary computing should be incredibly well placed for the future, notably in terms of job opportunities. Let us give examples of this.

Consider first the implementation relationship between computing and the other domains. In one direction this yields implementations of computing, but not merely the familiar varieties of electronic hardware. Babbage's difference engine was a 19th century purely mechanical computer. Today students can see implementations of computing across many domains. There are chemical computers, biological computers, DNA computers, and massively parallel social networks that compute. And there are oddities such as billiard ball computers.

In the other implementation direction, we see computational simulation, and its many uses in science, engineering, education, training, communication and entertainment. At the core of most games sits a simulated environment of some sort – whether of the physical, life or social worlds, or some combination of all three. And beyond simulation, there are deep and thought provoking scientific questions concerning whether computation underpins all of our physical reality, whether it is possible to create computational life, and whether artificial intelligence can yield human-level intelligence.

When we get to interaction, we see robots of all sorts, from traditional industrial robots, to self-driving cars and unmanned aerial vehicles, to rapid prototyping devices that can build everything from toys to buildings. We see robots in industrial, outdoor, and social environments. We also see an increasing variety of ways that computers interact with our bodies and minds. They are understanding what we say, interpreting our movements, and providing rich, interactive sensory and informational experiences. They are now starting to interface directly with our brains, providing the technological equivalent of mind reading.

Polyadic computing involves multiple domains and relationships. It includes prosthetic devices, caretaker robots, intelligent robots, mixed reality environments that combine the virtual and physical worlds with human interaction, and ubiquitous computing environments that embed networked computing throughout the physical, and potentially also the biological, world.

In summary, computing is the basis for a true science, and in fact for an entire scientific domain. When analyzed via the relational approach, it can be seen to span a wide array of important and exciting topics that hold promise for engaging and educating our students, and for helping them to prepare for their, and our, future.

Bibliography

Rosenbloom, P. S. A new framework for Computer Science and Engineering. *IEEE Computer*, 37: 31-36, 2004

Denning, P. J. & Rosenbloom, P. S. Computing: The fourth great domain of science. *Communications of the ACM*, 52: 27-29, 2009.

Rosenbloom, P. S. *On Computing: The Fourth Great Scientific Domain.* Cambridge, MA: MIT Press, 2012.

Denning, P. J. Opening statement, Symposium on 'Learning and teaching computing as science', *Ubiquity*, 2012.

Symposium on 'What is computation?', *Ubiquity*, 2010.