DOI:10.1145/1721654.1721657

Computing Paradigm Not a Branch of Science

EGINNING WITH THE headline, "Computing's Paradigm," The Profession of IT Viewpoint by Peter J. Denning and Peter A. Freeman (Dec. 2009) reflected some confusion with respect to Thomas Kuhn's notion of "paradigm" (a set of social and institutional norms that regulate "normal science" over a period of time). Paradigms, said Kuhn, are incommensurable but determined by the social discourse of the environment in which science develops.

The crux of the matter seems to be that computing can't be viewed as a branch of science since it doesn't deal with nature but with an artifact, namely the computer. For guidance, we reflect on at least one scientific antecedent thermodynamics, which originated from the need to understand the steam engine but is distinguished from steam engineering by its search for general principles, detached from a specific machine. The Carnot cycle and entropy theorem are scientific results, not feats of engineering.

The metatheoretical problem of computing seems mainly semiotic. Suppose, 200 years ago, somebody had created a discipline called, say, Thermozap, that included the study of the Carnot cycle and the building of new steam engines. Somebody might have come up with the insoluble problem of whether the new discipline was science or engineering. It was neither but rather a hodgepodge of things better left separated.

Computing is in a similar situation. There is an area (call it Knuth-Dijkstra computing) that studies scientific problems posed by the existence of computing devices. Thermodynamics was part of physics because steam engines use physical forces. Computing devices are formal machines, so Knuth-Dijkstra computing is a mathematical discipline. Then there is the computing discipline that builds systems (call it Denning-Freeman computing), which is definitely part of engineering. The error is in thinking they are the same. Both refer to the same device, generically called "computer," but is a misleading connection, since the two disciplines describe the computer in different ways—a formal model of computation in Knuth-Dijkstra computing, an actual machine in Denning-Freeman computing.

Denning and Freeman proposed a "framework" that takes the side of engineering computing (why I call it Denning-Freeman computing), describing development of an engineering system and leaving no doubt as to the envisioned nature of the discipline. All the purportedly different fields they proposed-from robotics to information processing in DNA-are actually different applications of the same paradigm. To consider them different would be like saying quantum physics is different for nuclear plants and for semiconductors. The physics is the same; what changes is the engineering process of its application, as in computing.

The abstract problem of symbol manipulation is mathematical and the subject of computing science. The instantiation of the symbol-manipulation model in useful systems is a problem for the engineering of computing, a discipline that is theoretically, methodologically, and conceptually separated from the mathematical study of symbol manipulation.

Simone Santini, Madrid, Spain

I wish to suggest ways to improve Peter J. Denning's and Peter A. Freeman's proposed computing paradigm in their Viewpoint "Computing's Paradigm" (Dec. 2009). While I accept the tentative five phases—initiation, conceptualization, realization, evaluation, and action—in the proposed paradigm, they are, in practice, incomplete.

While I agree with initiation (the existential argument followed by conceptualization) as the design argument, three additional phases are missing: The first is a phase 0 I call understanding (or problem understanding). Before one can pose the existential (Denning's and Freeman's initiation), a phase must address (problem) understanding, a key element in all complex computing domains. Moreover, understanding is associated with modeling, a key aspect of understanding. One cannot determine whether a system can be built or represented without the understanding needed to pose hypotheses, theses, or formal requirements. Understanding is often not addressed very well by beginning computing researchers and developers, especially as it pertains to information processes.

The second missing element of conceptualization is an explicit statement about bounded rationality, per Herbert Simon (http://en.wikipedia.org/ wiki/Bounded_rationality), a concept based on the fact that the rationality of individuals is limited by the information they possess, the cognitive limitations of their minds, and the finite amount of time they have to make decisions. Bounded rationality addresses the tentative nature of design and discovery as an evolving set of decisions posed against multiple criteria derived from understanding and initiation. The results from conceptualization, or design, must always be understood as both tentative and knowledge-limited.

Finally, a phase missing from evaluation and action is "technology readiness" (http://en.wikipedia.org/wiki/ Technology_readiness_level), especially in deploying real systems. A new technology, when first invented or conceptualized is not suitable for immediate application. It is instead usually subject to experimentation, refinement, and increasingly realistic contextual testing. When proven, it can be incorporated into a deployed system or subsystem. All information processes are realized and embedded within the context of existing deployed systems. Therefore, technology readiness of a posed information process must stand as a separate phase between evaluation and action.

1. Simon, H. Bounded Rationality and Organizational Learning. *Organization Science* 2, 1 (1991), 125–134.

2. Simon, H. A mechanism for social selection and successful altruism. *Science* 250, 4988 (1990), 1665–1668.

3. Simon, H. A behavioral model of rational choice. In *Models of Man, Social and Rational: Mathematical Essays on Rational Human Behavior in a Social Setting.* John Wiley & Sons, Inc., New York, 1957.

David C. Rine, Fairfax, VA

Authors' Response:

Our argument concerned computing's "belief system." Kuhn discussed belief systems in science. Whether or not we were true to Kuhn is irrelevant to our argument.

Santini says computing is about computers. We disagree. Computing is about information processes, and computers are machines that implement information processes. There are natural, as well as artificial, information processes. Computing is as much about computers as astronomy is about telescopes.

Computing does not separate neatly into math and engineering, as Santini claims. Computing increasingly employs experimental (scientific) methods to test hypotheses about complex information processes.

Santini's desire to parse computing into separate elements will fail, just as all such previous attempts have failed. Our collective concern with information processes keeps pulling all the elements together, no matter how hard we try to separate them.

Peter Denning, Monterey, CA **Peter Freeman**, Atlanta, GA

Hold the Accusations That Limit Scientific Innovation

I applaud the debate on MapReduce between "MapReduce and Parallel DBMSs: Friends or Foes?" by Michael Stonebraker et al. and "MapReduce: A Flexible Data Processing Tool" by Jeffrey Dean and Sanjay Ghemawat (Jan. 2010). But I strongly object to the former's criticism of the MapReduce designers, saying "Engineers should stand on the shoulders of those who went before, rather than on their toes." Creating an alternate method is not stepping on anyone's toes. Such accusations, besides being unjust, impede science.

Jonathan Grier, Lakewood, NJ

Authors' Response: As we noted in the article, the Map phase of a MapReduce computation is essentially a filter and a group-by operation in SQL, while the Reduce phase is largely a target-list computation in SQL. When user-defined functions are included in SQL (as they are in many commercial implementations), the functionality provided by parallel SQL DBMSs and MapReduce implementations appears to be the same.

The parallel DBMS literature, dating from the 1980s, includes hundreds of articles on implementation tactics. Our comment about "standing on the shoulders..." was meant to suggest that any new implementation effort should carefully review the prior literature to learn what past results are available, then add to the store of total knowledge.

The MapReduce team seemed not to have done this exercise. Hence the comment.

Michael Stonebraker, Daniel Abadi, David J. DeWitt, Sam Madden, Erik Paulson, Andrew Pavlo, Alexander Rasin, Cambridge, MA

Even in the Classroom, a Click Is Just a Click

The news item "Web Used for Final Exams in Denmark" (Jan. 2010) gave the impression that such an approach was never tried before. I have taught computer- and network-security-related classes for the past eight years, incorporating the Internet as a tool students use during class, including on quizzes and exams. I am sure I am not the only instructor in the U.S. allowing students to use the Internet for research and comprehension in the classroom.

Is Europe just now discovering the value of Internet searches in education? There is no reason to require that students memorize details accessible at the click of a mouse, when they might better spend their time analyzing and comprehending. The old way of requiring that students memorize facts from textbooks should give way to methods of learning more in tune with the Y generation.

Moreover, exams should be tailored so students don't just regurgitate facts, but make facts accessible over the Internet, then require students show they have comprehended them to solve problems. This new paradigm in testing emphasizes comprehension over memorization. **Bela Erdelyi**, Lincroft, NJ

How to Honor the Heroes of CS

Communications cover article "Amir Pnueli Ahead of His Time" (Jan. 2010) mourned the passing of Amin Pnueli in November 2009. Likewise, Communications mourned (Nov. 2008), along with the rest of the computer science community, the disappearance and passing of Jim Gray. Tragic as these events are, they are sure to be followed by others, as computer science is no longer in its infancy but well past middle age. I see the risk that Communications covers (and articles) could turn into a gallery of the revered heroes of our science who will be passing away in ever greater numbers. Communications could instead honor its icons by, perhaps, adding an obituary column, even as a permanent feature.

Panos Louridas, Athens, Greece

Editor's Response:

Communications does indeed publish obituaries to note the passing of prominent computer scientists. In certain cases, however, the Editorial Board deems the event to be deserving of further recognition. Jim Gray was in full vigor when he disappeared without a trace in January 2007, as was Amir Pnueli when he passed away in November 2009. In both cases there was a sense of unusual or unexpected tragedy, which explains the degree of coverage in Communications.

Corrections

In the article "Amir Pnueli: Ahead of His Time" (Jan. 2010), it was reported that Pnueli was born in Nahalal, Israel, in 1941. The State of Israel was yet to be declared in 1941. Nahalal was then in the British Mandate of Palestine.

The article also noted that Pnueli worked with David Harel on Statecharts. The Statecharts formalism was developed by Harel. Pnueli was involved in the development of Statemate, a software system implementing the Statecharts formalism.

Communications welcomes your opinion. To submit a Letter to the Editor, please limit your comments to 500 words or less and send to letters@cacm.acm.org.

© 2010 ACM 0001-0782/10/0400 \$10.00