Through a Glass, Darkly
A Quibble About “Computation”

Robert L. Glass

It has become popular in computer science circles in recent years to use the word “computation” to describe all the things that computers do. And, to be honest, that bugs me.

Look at the computing applications in our IS/IT fields. It’s not an accident that the dominant word in our field is not “Computation,” it’s “Information” (it’s the “I” in IS and IT, of course). In fact, computers in our field rarely do “computation;” instead, they process information (Accounting applications, of course, do a bit of computation, but it’s mostly about “arithmetic,” hardly what those computer scientists are thinking about when they use the term “computation”).

It has been several decades since one of our professional societies, the Association for Computing Machinery, shortened its name to “ACM,” for a couple of very good reasons. For one, the ACM had long since ceased to be about “machinery” (software, one of the key subjects the ACM deals with, has nothing to do with machinery). For another, and more to the point, most computing hardware is used for things much more profound these days than “computing.” Nearly all applications these days are about processing some kind of information – business information, software tools information, games information, ... The only application domain that is primarily about computing things is the scientific/engineering domain. And that represents a fairly small percentage of the total uses of computers.

So the notion that “computers” (read, preferably, “information processors”) deal with “computation” is, at least in my mind, pretty obsolete. But perhaps I’m wrong. Is there some broader context in which the term is still useful? That’s a question for a dictionary to decide, of course. So I Googled “define computation,” and this is what I got:

- [calculation](#), **computation**, [computing](#) (the procedure of calculating; determining something by mathematical or logical methods)
- [calculation](#), **computation**, [figuring](#), [reckoning](#) (problem solving that involves numbers or quantities)

(My large and heavy dictionary, which I access when all other attempts to get a lighter-weight (literally) definition have failed, tells me something fairly similar, except it uses the term “computer” heavily in its definition).

Now there's a certain amount of ambiguity in those twin definitions. One out of four of them, "logical methods," could actually encompass what we do to/with information. But even more importantly, I think, three out of four of them don't. Mathematical methods are about symbol manipulation or numeric manipulation, not information manipulation. "Numbers or quantities" are a tiny part of the information that our current-day applications manipulate. With these definitions, "computation" is clearly on the borderline of what our current-day computers are tasked to do, but hardly in the mainstream.

So why am I engaging in this rant?

From time to time, someone well-known in the computing field brings up the subject of "computation," implying that is the essence of what computers do (I suppose my real argument here is about the use of the term "computer," but "computation" is what most of these well-known people are pushing these days).

Here's one example. Some key computer science academics are suggesting that a course in "computation" should be in the core curriculum universities everywhere offer. Under that umbrella, their intention is to emphasize the use of computers and, of course, computation, for problem solution. And they see those computation courses having a heavy computer science and mathematical flavor.

Here's another example. One of those news summary emailings, the kind that report on matters of interest to a particular professional group (in this case, it was the ACM's service), referenced an article on a new research project at England's University of Sussex designed to do language understanding analysis. Neat, and interesting, and at this point not particularly relevant to my rant.

But the lead sentence of the article said those researchers are asking the question "can human computation solve computer language comprehension?" And I found myself wondering, yet again, what this issue had to do with computation, by humans or computers or anyone/anything else.

The article went on to talk about something called a "Phrase detective," and a data-gathering research method that involved gathering human information on parsing language content in order to provide the computer with a basis for making its own decisions. Fascinating, a worthy subject for research, and one for which I wish those researchers well.

But this is hardly about "computation." In fact, there was a telling sentence buried deep in the article that said "Players in the game are helping to create a resource that is rich in linguistic information..." I paused over that sentence, with some sense of glee. The overall article may think it's about computation, but deep in its logical innards, it knows that it's really working with information.

Now I suppose I have to admit that this is a fairly trivial rant. Who cares, in an era where the word “computer” is commonly accepted to describe those information processing devices that only occasionally compute, whether we then attribute the word “computation” to the task in which they engage? Do you?

“Through a Glass, Darkly,” is a Biblical expression for the unclear way in which we see the world around us.

Robert L. Glass (rlglass@acm.org) is president of Computing Trends, publishers of The Software Practitioner newsletter, and an Honorary Professor of Software Engineering at Griffith University, Brisbane, Australia. He has been active in the field of computing and software for over 50 years, largely in industry (1954-1982 and 1988-2005), but also as an academic (1982-1988 and 2005-present). He is the author of over 25 books and 95 papers on computing subjects, Editor of The Software Practitioner newsletter, Editor Emeritus of Elsevier’s Journal of Systems and Software, and is a former columnist for several periodicals including Communications of the ACM (“The Practical Programmer”) and IEEE Software’s “The Loyal Opposition.” He was a Lecturer for the ACM for 15 years, and was named a Fellow of the ACM in 1998. He received an honorary Ph.D. from Linkoping University in Sweden in 1995. He describes himself by saying “my head is in the academic area of computing, but my heart is in its practice.”