

The end of Computing Science?

“The price of reliability is the pursuit of the utmost simplicity. It is a price which the very rich find most hard to pay.”

Sir Antony Hoare, 1980 .

In academia, in industry, and in the commercial world, there is a widespread belief that Computing Science as such has been all but completed and that, consequently, computing has “matured” from a theoretical topic for the scientists to a practical issue for the engineers, the managers and the entrepreneurs, i.e., mostly people — and there are many of those! — who can accept the application of science for the obvious benefits, but feel rather uncomfortable with its creation because they don't understand what the doing of research, with its intangible goals and its uncertain rewards, entails. This widespread belief, however, is only correct if we identify the goals of Computing Science with what has been accomplished and forget those goals that we failed to

reach, even if they are too important to be ignored.

I would therefore like to posit that computing's central challenge, viz. "How not to make a mess of it", has not been met. On the contrary, most of our systems are much more complicated than can be considered healthy, and are too messy and chaotic to be used in comfort and confidence. The average customer of the computing industry has been served so poorly that he expects his system to crash all the time, and we witness a massive worldwide distribution of bug-ridden software for which we should be deeply ashamed.

For us scientists it is very tempting to blame the lack of education of the average engineer, the short-sightedness of the managers and the malice of the entrepreneurs for this sorry state of affairs, but that won't do. You see, while we all know that unmastered complexity is at the root of the misery, we do not know what degree of simplicity can be obtained, nor to what extent the intrinsic complexity of the whole design has to show up in the interfaces.

We simply do not know yet the limits of disentanglement. We do not know yet whether intrinsic intricacy can be distinguished from accidental intricacy. We do not know yet whether trade-offs will be possible. We do not know yet whether we can invent for intricacy a meaningful concept about which we can prove theorems that help. To put it bluntly, we simply do not know yet what we should be talking about, but that should not worry us, for it just illustrates what was meant by "intangible goals and uncertain rewards".

And this was only an example. The moral is that whether Computing Science is finished will primarily depend on our courage and our imagination.

Austin, 19 November 2000

[Written for the Communications of the ACM]

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
USA