A few comments on "Computing as a Discipline"

The following remarks have been triggered by the report "Computing as a Discipline by Peter J. Denning (Chairman), Douglas E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, and Paul R. Young. (see CACM, Jan. 1989, Vol. 32, No 1, pp. 9-23. Criticisms from my side should not be interpreted as my judgement that, under the circumstances, the authors could have done much better: they had to please each other and there was politically no point in departing too far from the status quo.

I like its title, both for containing the word "computing" rather than "computer" and for containing the word "discipline" (which is dear to my heart). Regrettably, its title is the part I like best.

My doubts already start at the first question it poses: "Is computer science a science?" From my strictly personal point of view, that question has been answered a long time ago in the following sense: I have now been involved for more than 37 years, at least 80% of my professional activities were related to automatic computing, and I have never doubted that I was a scientist. So, what's the fuss about? For me, the fuss can mean only one thing, viz. that in the past we have been too liberal in applying the name "computer science": under its auspices we have allowed all sorts of activities that are scientifically not up to par.

Such a conclusion, of course, invites the proposal to prune the junk away; personally I would whole-heartedly support such a proposal, but -regrettably and understandably- the task force did not make it. It came, instead, with a picture in which computer science is mathematics, natural science, and engineering, all rolled into one.

But even when I take into account that all important words (such as mathematics, natural science, engineering, university, and education) subtly change their meaning when one crosses an ocean, I find their synthesis most unconvincing. They give paradigms for work in mathematics, in natural science, and in engineering, but I don't recognize them at all! That is, even if we take the width of oceans into account, a bit alarming, for I worked in all three areas. (For your information: my degree is in theoretical physics, the degree of my students in Eindhoven was that of "mathematical engineer".)

At least as revealing as the absurdity of the three four-step paradigms are the sentences by which they are followed: each of them, the mathematician, the scientist, and the engineer, "expects to iterate these steps". Not necessarily so! There is no reason why one should not try and later learn to expect to be almost always right the first time. It is van Beethoven, who was a very iterative, almost painful composer, versus Handel, Haydn, Mozart & Schubert. I encountered the cult of iterative design for the first time when I came in contact with Anglo-Saxon colleagues; later I learned how deeply ingrained it is in the Anglo-Saxon character: specifically in the Anglo-Saxon world, the suggestion to take not van Beethoven but rather the other four as your rôle model has, over and over again, been rejected as an indecently presumptuous proposal. In short: the picture the task force presents is parochial and distorted.

Consequently, their plea for an "experimental" component of computing science remained, as far as I am concerned, unconvincing. Nor did they succeed in making me believe in the desirability of computing laboratories for educational purposes, despite sentences such as "[Laboratories] should provide concrete experiences that help students understand abstract concepts." Everybody knows that programming is most easily and effectively taught to students that have never touched a machine and are not given access to a computer as long as the course lasts. The explanation is simple: for them it is much easier to ignore that a program admits the interpretation of executable code and thus to part from operational reasoning. And as far as "understanding abstract concepts" is concerned, you should help your students to understand them in their own right, and not in terms of a (computational) model a laboratory confronts the students with. Because departmental investments act as status symbol, it is probably politically suicidal to suggest that most educational equipment be kicked out; I guess the task force had no choice but to concoct a rationale for the goodies.

For the sake of completeness I mention that the report is rather silent on the lack of coherence displayed today: cognitive psychology has as much to do with formal logic as bacteriology with the study of music.

They are right in decrying the current level of introductory programming courses that turn away the best students, who want a greater challenge. They include, however, in their proposal "walk-through" and "hand simulation"; this inclusion hardly suggests that the programming challenge be faced more seriously.

* * *

In one respect, I think, the report is fundamentally flawed. They present computing as the synthesizing continuation of established disciplines such as mathematics, natural sciences, and engineering, as the result of a smooth evolution. They stress computing's structural similarity to those older disciplines.

I think that they are profoundly mistaken, that the automatic computer embodies such a radical novelty that the analogies between computing science and the older disciplines are too shallow to be of any help. It is great to be in computing science, and not because it is just like mathematics, natural science, and engineering, but because it is radically different. It is so fascinating because the computer's existence has faced us with an intellectual challenge without even the vaguest precedent in our cultural history. None of the scope of this challenge is reflected in the report, and that is what makes the report so bland. But, please, don't blame the authors. You cannot ask a task force to spell out a truth that is too disturbing.

Austin, 13 February 1989

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