Computing the future?

I read almost all of "Computing the future: a broader agenda for computer science and engineering", edited by Juris Hartmanis and Herbert Lin, for the National Research Council. The book is printed with soy ink on acid-free recycled stock (and that is perhaps the best we can say about it).

One way of writing a book by a committee is to restrict its contents to the intersection of the various opinions; it is a technique that tends to lead to short texts of high quality, and has the advandage that in the case of a sufficiently diverse committee, it leads to no text at all. The current book seems more to have been compiled by the dual technique that includes the union of the various opinions: it is a fat report --100,000 words-- of uneven quality and --as is only to be expected under such circumstances-- most of it of dubious nature.

It is depressing literature because major parts of it have been written with so little competence, so little idealism, and so little respect for its readers. For me personally, it was particularly disturbing as it challenged the wisdom of my decision to emigrate to the USA: it mostly offers recommendations for mediocrity and worse, urging the American CS departments to strengthen their worst characteristics.

Compared to the European Universities in whose climate I grew up, the American University is much less isolated from the society surrounding it. I knew this before I came, I observe here the experiment (sometimes bewildered!) with interest and conclude that the US campus is insufficiently protected against outside pressures and fads. But The Book recommends to strengthen the ties with the computing industry even further. In view of the divergent goals, the cooperation is, of course, doomed: the main purpose of industry is to make money and, for the sake of its continuity and stability, industry tries to do that in a way that makes it as independent as possible of the special competence of individual employees, whereas the primary purpose of the university is to lay for its students the basis for an intellectually rewarding life.

My former responsibilities for Dutch academic Computing Science implied that I had to study what made tentative academic disciplines viable and what constraints were indispensable for the creation of a first-class department.

One requirement that stood out was coherence of the subject matter: in the knowledge and abilities to be transmitted to the next generation, the knowledge should enable to improve one's abilities, and the application of the abilities should improve one's knowledge. Consequently, it was decided that "applications" were not part of computing science: since computers are truly "general purpose equipment", inclusion of computer applications would destroy the coherence and thereby fatally hurt CS as a scientific discipline. It was the right decision, and it was possible to take it by delaying the creation of separate Departments of Computing Science until the nature of its intellectual contents stood out with enough clarity. The American Universities were less fortunate and they were forced to start prematurely: for lack of alternative, they started with cocktail-departments, many of which --in spite of efforts towards intellectual consolidation-- still lack the required coherence. And what does The Book recommend? To "broaden the agenda", to embrace all sorts of applications in interdisciplinary efforts! Years ago, I have learned to interpret pleas for interdisciplinary science as symptoms of anti-intellectualism, and I found the recommendation to "broaden the agenda" in the way proposed just sickening.

A second requirement for the viability of a tentative scientific discipline is the "staying power" of its subject matter: material whose relevance has a half-life of five years is better ignored than taught. (The European conclusion was to exclude, in view of the fickleness of the market place, industrial products from the curriculum.) Here, The Book is mostly ridiculous. It recommends to support CS&E as "a laboratory discipline (i.e. with both theoretical and experimental components)", while admitting that upgrade is "especially important on systems that retain their cutting edge for just a few years". (The Book complains about the difficulties of working in such a fast-changing discipline, but it only changes fast if one fails to ignore the --evidently-- irrelevant.)

The Book is quite explicit --p.154-- : "In the fast-changing CS&E environment, laboratories must be completely revised frequently, i.e. every several years.", but I expect the trained politician to become very suspicious that the stress on the supposedly experimental nature of CS&E is primarily a political device to extort money, for, moreover, the argumentation of why CS&E should be viewed as "a laboratory discipline" is, of necessity, as weak as the suggestion is dubious: the only argument in favour I found on p.26 "Many operating systems [...] such

as MS-DOS and Unix make use of many years of experimental CS&E.". The suggestion --p.149-- that the traditional quality criteria for promotion and tenure have to be revised for "CS&E experimentalists" should make the trained politician extra suspicious.

To quote a few offensive/ridiculous sentences:

- academic CS&E must increase the number of applications of computing [...] in areas of economic, commercial, and social significance --p.141-- .
- Research can thus be viewed as providing a "service function" to those who develop the nation's computing capability, i.e. the product developers. --p.40--
- Intellectually, CS&E includes programming, which allows a programmer's thoughts to spring to life on the screen. --p.213--
- Abstraction is a generic technique that allows the human scientist or engineer to focus on certain features of an object or artifact while hiding the others. --p.169--

Let me quote, in contrast, one of the most sensible sentences I found, tucked away on p.132: "Efforts should be made to reduce the distance between theory and practice, to develop researchers who can do both."

Fortunately, the Appendix "Contributors to Computing the Future" mentions only one faculty member of UT Austin.

Austin 15 October 1992

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