

Trip report E.W.Dijkstra, Munich - London, 16 - 29 September 1979.

Actually these were two trips, between which I was three days at home. The first trip was to the 4th International Conference on Software Engineering in Munich, the second one was to Euro IFIP 79, held in London. I visited them both because for both conferences I had agreed to deliver an invited speech at the closing session. For the first conference I travelled by train, for the second one I did so by plane; during the first conference I stayed in the little village of Holzkirchen as the guest of Professor Gerhard Seegmüller, during the second conference I stayed in the Wembley Eurocrest Hotel. But apart from such minor differences the two conferences were sufficiently similar to be dealt with in a single trip report. Both attracted slightly over 1000 participants --and, hence, were regarded by their organizers as a success-- and both had the same sort of charter. I quote from the program of the 4th ICSE:

"The tone and emphasis of the present Conference has clearly been set by the fact that interest and emphasis is currently focussed on the tooling and tool support of the programming and software maintenance process; the use of computers to produce and maintain higher quality, timely, cost-effective and long-life software."

I quote from the Proceedings of Euro IFIP 79 --with the conference subtitle "European Conference on Applied Information Technology"--

"The whole emphasis is on applications, in government, in control of the environment, in the expansion of human capability, on social effects."

Both conferences pandered to current fashions and tastes of the many, as true products of bestsellerdom. The organizers sold their products and seemed content, but what went over the counter was in general just as vulgar as was only to be expected. I shouldn't have given in when their organizers invited me, I shall not attend such conferences for years to come, for they are a depressing waste of time.

The one in Munich depressed me deeply. (That is not to say that the one in London was better: presumably my experiences at the first conference had suitably lowered my expectations of the second one.)

In many fields the traditional controversy between "theory" and "practice" may be a very real and unavoidable one. What fascinates me in computing is that there the controversy need not exist, for what is theoretically sound and scientifically elegant is usually at the same time eminently applicable in practice. This has been expressed by many --see the chapter "Theory and Practice" in "Software Engineering Techniques", Rome, 1969, Eds. J.N.Buxton and B.Randell-- and experience has given abundant confirmation of it. And yet, at conferences like these, this stale and obsolete notion of the controversy is still in full vigour! The mathematician is still depicted as someone walking with his head in the clouds, the practitioner as someone who stirs in the mud (or even: has to stir in the mud) but does the "real work", etc. Hackneyed phrases seemed to be the dominant mode of thought, and, as a result, I felt in a primitive society indeed. This feeling was enforced by the observation that exposure of the clichés, as well as challenge of the tacit assumptions, was strictly taboo. This was even the case with some explicitly stated articles of faith, such as

Because modern technology can produce more chips than we need, computing science's most important task is to discover new applications for them.

Software engineering is in such a bad shape because it has refused to automate its own job.

Needless to say, I found these articles of faith somewhat hard to subscribe.

\*            \*            \*

On one of the first days in Munich I was vividly reminded of my third visit to the USA, when I attended the Operating Systems Principles conference in Gatlinburg, Tennessee (1967). To my utter amazement the poor quality of the then current operating systems was there blamed on the fact "that we did not have a Language for writing Operating Systems in"; as a result of the predictably fruitless and confused discussion that then followed, the absence of a "Specification Language" was invented to share the blame.

I was reminded of that incident when I was approached by three young Americans that (on account of a remark made while lecturing at the University

of Texas at Austin) wanted to differ of opinion with me about the design process. Eventually I had skirted the (in their minds important) issue of "what language to use". (Shortly after my discovery that I was talking to three psychologists, Susan L.Gerhart saved me.)

This happened after I had listened that day to two invited American speakers --Dr.Barry Boehm and Dr.Charles Vick-- who had shocked me by their extremely sloppy use of their own language: full of meaningless jargon and awful neologisms --from Boehm I learned that besides "human factors" and "human engineering" (terms I cannot hear without a shudder) we now have "software psychology"-- and, in the case of Vick, presented as if logic and grammar were dispensable commodities. Then I suddenly realized that both did business or belonged to the DoD, remembered how profoundly inadequately English had been used in the Ironman and Steelman requirements. Since that moment I cannot get rid of the lurking suspicion that much of the American preoccupation with all sorts artificial special-purpose "languages" is closely related to the fact that large layers of that society --the academic world included-- is basically illiterate. Such an assumption of wide-spread illiteracy is frightening, and has worried me all through the two conferences. We may have been unlucky with the selection of the overseas speakers, but as far as I can remember Professor E.J.McCluskey from Stanford has been the only American speaker I heard that did not speak undiluted nonsense most of the time. After Boehm's talk in London I asked him, because he had been allocating software costs and I had missed the item, how much of the costs he felt could be attributed to the use of sloppy language. His answer was noticeable in two respect. The one is that he (almost automatically, it seemed) narrowed "sloppy language" down to the use of undefined terms; the other is his actual estimates "In small student projects I would say 20 per cent. and in large projects much more." The result is that the lurking suspicion is still with me; and because the U.S.A. is a large and influential country I have now and then a hard job at not coming too close to despair. One thing is certain, however: independently of the question whether the lurking suspicion is justified or not, the intrinsic problems of computing science should not be confused with difficulties that have been created by the American educational system.

\* \* \*

In Albany and in Santa Cruz I had had two rehearsals for my invited speech; in Albany it had fallen flat on its face, in Santa Cruz --but that was for an audience that had been educated during the preceding two or three weeks-- it had been a great success. In Munich, where I addressed about 1000 people at the end of the third and last day, it was a mild disaster. In London I therefore used the time I had available in my hotel room to reconsider its presentation. I gave it at the end of the fourth and last day, and this time I seemed to reach again most of my audience (with the exception of the Chairman of the Organizing Committee, who used the dimly lit Grand Hall to fall soundly asleep on the front row). The organizers had scheduled me for the closing session with the hope of keeping the participants' attention to the very last moment, but that gimmick failed: the last day only a fifth of the participants turned up. (I couldn't blame them.)

In the first week I had enjoyed the beauty of the countryside of Southern Bavaria; after Euro IFIP 79 I stayed the last night with Mike Woodger, with whom I had a very nice stroll amidst the Surrey hills on Saturday morning. My KLM-flight back home was half an hour late.

Plataanstraat 5  
5671 AL Nuenen  
The Netherlands

prof.dr.Edsger W.Dijkstra  
Burroughs Research Fellow