Subject: Will the Real SE Metaphors Please Stand Up!
(or, I Never Metaphor I didn't like!)

Dear Dr Neumann:

I do not disagree with Dijkstra (as quoted by Horning (SEN,10(3)p.2)) that: "the true subject matter of the programmer is not programs, but the design of computations." (See also, EWD227.) However, I find Dijkstra's homily (and thus Horning's suggestion that we become "computation engineers") to be maddeningly parochial. I favor instead, the following: a program is merely a script for a (family of) process(es). Stay tuned and I will tell you why I prefer it.

It is its wide (and ever-increasing) scope of application that makes computers (and their servants, the software engineers) difficult to catagorize. The computer is by far the most versatile and challenging metamedium ever devised./3/ It has already fascinated millions of people, including: brilliant mathematicians (eg E. W. Dijkstra, C. A. R. Hoare), gifted engineers (eg Alan Kay, Raymond Kurzweil) and inspired entrepreneurs (eg Steve Jobs, Mitchell Kapor). In their ubiquity, the computer, and his faithful companion, computer software, have outgrown the computational metaphor.

I hope that there will always be room under the umbrella of software engineering for those who are entranced by the computational metaphor of computing, but they are destined to remain in the minority. Meanwhile, the explosive growth in computer use is being served by software engineers who are captivated by the simulation (or modeling) metaphor of computing./3/ (I am not proposing that we let a thousand metaphors bloom, only claiming that one is insufficient; especially if it is the computational metaphor.) Now, "1-2-3" and "WordStar" are examples of software created within the simulation metaphor. In addition, they are both products, and much of what the engineering community knows about product (re)development is applicable to them. Doesn't anyone remember where Software Configuration Management came from? (See SEN,10(1)p.13 for an (attempted) elucidation of this point of view.)

For mathematicians (and, therefore, many computer scientists) the problem with the term "engineering" seems to be that it invokes images of a smelly, unsightly, merely approximate world in which Murphy's Law holds sway. Thus mathematicians (and, therefore, many computer scientists) prefer to inhabit abstract, non-threatening, fantasy worlds in which Murphy's Law is inoperable./2/ (This is no disparagement of mathematicians, in fact it has been argued that they are only more concerned about preserving their sanity than the rest of us! Besides, I hold a university degree in mathematics.)

Nonetheless, one of the many dangers that Star Wars poses is that it may be realized by people who see themselves as computer scientists (ie mathematicians, albeit once removed) rather than by people who see themselves as (eg aerospace) engineers. Either way, Star Wars could

Letter: Editor, SEN, from D. A. Nelson, "SE Metaphors" (16.09.85) (2 of 2)

bring an end to my favorite world. However, I would rather risk being blasted into eternity by the failures of committed engineers (employed by the low bidder) than by the fantasies of those who profess that there is substantive knowledge in (eg predicate) logic! Thus I believe that adherence to the paradigms of engineering in developing software for use in the real world is not only desirable but necessary (although not sufficient) for our survival. (Bill Riddle's objections (SEN,10(2) pp.1-3) to the contrary notwithstanding.)

As F. L. Bauer has said: "The computer, one of the greatest inventions of engineers, has to go the complete way of engineering to its end."/1/ Lest that be interpreted as a statement about computer hardware, I remind you that Bauer was among those who coined the term "software engineering" and he was also chairman of the first conference to use that term in its title./4/

Sincerely,

Dave Nelson

David A. Nelson
Software Engineer

DAVID A. NELSON
Information Engineering*
Elisenweg 12

8100 Garmisch-Partenkirchen
Federal Republic, Germany
Tel: (49) 8821/51038

References:

/1/ Bauer, F. L., "Software Engineering," (pp. I-267 - I-274) in "IFIP Congress 1971," North-Holland (1971).

/2/ Kay, A., "Inventing the Future," in Winston, P. & Prendergast, K. A. (eds.), "The AI Business," MIT Press (1984).

/3/ Kay, A., "Computer Software," Scientific American, 251(3)52-59(Sep. 1984).

/4/ Naur, P. & Randell, B. (eds.), "Software Engineering" (Report on a conference sponsored by the NATO Science Committee, Garmisch, October 1968) Scientific Affairs Division, NATO, Brussels (Jan. 1969).

16 September 1985

* formerly Data Structuring