Richard punctures a windbag*

Bernard Sufrin

for Richard Bornat, Aldeburgh, 9th June 2009

It's a great pleasure to be here today as one of Richard's many friends and collaborators. I envy his capacity to continue to contribute to Computing Science.

Richard has contributed a lot to several fields; and he has worked in very many fields. A back-of-the-envelope list I just made included: Compiler-writing, Language Design, AI (tell people about the heterarchy conjecture, Richard!), Editor design, HCI, Proof. He's a great teacher and a great learner, and he can also think on his feet.

And Richard is also a contrarian. he taught -- or at least, inspired -- me to be a contrarian, too; and (with a few thousand colleagues at Oxford) I have had good reason to be grateful for that in recent years.

Before I tell you this short tale about Richard, I have to warn you that, although we have been friends and colleagues for nearly 40 years, he thinks I am an habitual liar; so he will almost certainly try to convince you that it is a *tall* tale.

I first met Richard and Joanna in 1972 when Richard moved to Essex University from Manchester. I remember the afternoon we met -- it's etched on my mind. It was his first day in the Computing Science department.

Computing Science (or Comput*er* Science as the Americans called it) was then in its infancy; or perhaps I mean its childhood. Either way, it was certainly not mature! And it was a bit like the wild-west: as well as its farmers it also had its bandits, and its rustlers, and its snake-oil salesmen.

The head of our Department of Computing Science, Tony Brooker, was a brilliant and unpretentious polymath who had built relay computers at Imperial College, before going to Cambridge, and then Manchester where he worked with Turing,

*An essay in nanobiographics.

designed the first «autocodes» (programming languages that predated Fortran), designed the first «compiler-compiler», and made the throw-away-remark that led to Atlas being designed with 128 integer index registers. Richard and I both owe him a lot I think.

Brooker used to say that there were two fundamentally different approaches toward programming, and used to describe their adherents as either «primitives» or «space-cadets».[1] The primitives were content for programmers to code directly in assembly-language (or octal), while the space cadets believed in high-level languages and compilers.

There were certainly a lot of primitives at Essex. For them software was a way of opening and closing logic gates; and the closer to the machine instructions they worked, the more comfortable they felt.

Before he succeeded Tony as head of our Department, the chief primitive ran a research project that had set out to build a medical informatics system suitable for use by GPs. As a matter of principle they had decided to build it in Honeywell 316 assembly language, and to build their own operating system. They kept telling us we were crazy for suggesting that they write it in a high-level language. We even offered to write a BCPL compiler for them; we had already written two and it might have taken us about 3 weeks. Eventually the 316 obsolesced, and the project was dead.[2]

But enough about primitives!

Richard and Pat Hayes and I had three adjacent rooms on corridor «4B».[3] This was the corridor where the «space cadets» lived -- I think Mike Brady was there too, and Lockwood Morris co-reinventor[4] of the notion of «continuation» had just left. The beastlier of the primitives used to call it «horridor 4B», or just «the horridor».

For the most part the space cadets bonded naturally and easily since we were mostly young and mostly leftish or avant-garde in political and technical outlook; we were into language design and semantics, and we felt very much like we were disapproved of, or worse, by the rest

[1]Donald Michie reported this in an essay in one of the «Machine Intelligence» volumes, but I forget which one.

[2]Several years later when I returned to Essex from the US where I had contributed in small way to building the second generation ARPANET, the same chief primitive told me that I was off my rocker if I thought there was any future in local packet switching networks of heterogeneous machines. Mind you, even some of the space cadets were sceptical: they labelled my idea a «Thinkatron». Maybe Richard remembers this.

[3]Bruce Anderson arrived a couple of years later.

[4]The essence of the continuation idea was Peter Landin's, and had been embodied in his J operator. Sadly Peter died last week.

of the department.

Across the corridor from Richard, Pat, and me was John Laski. If you know of him at all these days it is probably as co-designer of an early simulation language called CSL.

He was certainly a space cadet. But to say that he spoke with a posh accent is to euphemize mightily. His diction was close to incomprehensible to anybody whose childhood had been spent anywhere further north than Kings College, Cambridge, or further west than Kensington High Street.

Those who managed to penetrate the superficial incomprehensibility of John's diction were rewarded --- by exposure to the fundamental incomprehensibility of his discourse. And to say that John's discourse was pretentious and obscure would be to compound the euphemism.

This combination of diction and discourse made John seem arrogant. Actually it would be more accurate to say that it did nothing to conceal John's actual arrogance. As Reader in Computing Science he outranked us academically, and tried to conduct himself as if he were Chief Space-Cadet.

Of course one can be prescient and correct despite being pretentious, arrogant, and obscure: it just makes it harder for people to learn from you.

For example, John published an article in *Machine Intelligence* #3 in 1968. It was called «The Morphology of Prex -- an essay in meta-algorithmics». In retrospect it seems to have been about abstract data types and their representations. It could have been enormously influential.... but it was written in Laskispeak; so it wasn't.

Now we all know how Richard hates pretention and obfuscation; and in retrospect I suppose it was inevitable that Laski would rub Richard up the wrong way when they met. On the afternoon in question, Richard, Laski and I were sitting in Richard's room, talking about programming languages. The great man (I mean Laski) was holding forth about something-or-other related to the semantics of Algol.

Since Richard had only recently written an Algol compiler, he also knew something about the semantics of Algol -- but from a practical perspective. A perspective that was evidently far too practical for Laski -- who soon decided that Richard must be a «crypto-primitive» and started baiting him with not understanding the lambda calculus, not understanding the Scott-Strachey approach, not understanding continuous functions, ...

The more Richard tried to explain his position to Laski, the more pretentious and abstruse Laski became. Before too long

Richard was raising his volume (some of you may have encountered this over the years), and Laski was raising his pitch. Neither were going to give ground, and neither were going to be beaten.

All at once, at Richard's turn, there was a pause.... Laski waited.... I waited.... silence.

But Richard's ability to think on his feet suddenly returned; he leapt up, shouted «Oh Fuck Off John»; then strode from the room and slammed the door.

In the 37 years we have been friends I've heard him tell quite a few people to fuck off. And often it has been me. And sometimes it has irritated or bewildered me. But I've learned to think of it as Bornat's revision of Wittgenstein's dictum -- something like: «Whereof I cannot speak, thereof I must say Fuck Off!»

Seriously: I'm proud to have been his friend for so long. And if (after this workshop) I'm still his friend then so much the better.

Any questions?

Now let's do some mathematics!