Extract from

PROFESSOR CLEVERBYTE'S VISIT TO HEAVEN

N. Wirth, ETH Zürich

Abstract

The following fable is a grotesque extrapolation of past and current trends in the design of computer hardware and software. It is intended to raise the uncomfortable question whether these trends signify real progress or not, and suggests that there may exist sensible Limits of Growth for software too.

When I had been dead for several weeks, I began to get a little anxious. I had been hovering around, first experimenting with my novel facilities and freedom from all earthly limitations. Perhaps I ought to mention at this point that I had been a manager of a software house, and my decease had been a direct consequence of our decision to introduce both a new programming language and a new operating system at the same time. The ensuing difficulties were enormous and responsible for my spending the rest of my life on the job.

So I was disappointed to see how little difference my absence made, in spite of the fact that I had been the only one intimately familiar with all the details of these new systems. I realized that a little more or less confusion didn't really matter.

Hence I became anxious to direct my course upwards. Fortunately I remembered the report of Mark Twain's Captain Stormfield , and therefore was neither surprised by my exhilarating rush through space, nor did I expect to enter a heaven of eternal bliss. But

I expected that it would be a place of unlimited opportunities where nothing was impocsible. This expectation is, of course, quite typical of a man from the software profession.

Heaven is a complex place, and it is also astonishingly modern; I was taken aback to discover large boards with light-displays and computer terminals used to find your present location as well as the shortest path to any desired location or department. The boards list all possible subjects you may think of. They continually expand as new departments with imaginative names emerge, one about every second. I readily found Software Engineering - merely the o had been misspelled as an a, perhaps by a German clerk - and I headed off in its direction. As a new department, it was located at heaven's periphery, and I marched for several days.

When I finally reached my blessed destination with sore feet, I found the quarters almost deserted. But as luck would have it, shortly thereafter I spotted a man carrying a deck of punched cards. I was overjoyed when I recognised him as my old friend Jonathan Flagbit who several years ago had switched from computing to life insurance. "You here, inspite of all!" I exclaimed; "You don't seem to have kept up with progress" I sneered referring to his card deck.

"Don't jump to rash conclusions, Cleverbyte, I've gone through all the stages up here, and we've got he most modern equipment you haven't even dreamt of".

Being quite excited at this prospect, I asked: "May I see your modern equipment?"

"Of course you may, everything is possible up in heaven and even more so in the Software Department ofer there. All you need is to make a wish, and it shall be fulfilled".

I told him grudgingly that I could have spared my sore feet had I known this beforehand, and he replied:

"Every newcomer indulges in wishing, but soon they get tired of it. It's deceiving in the long run. Too often there are small bugs, and you get something different. So wishing isn't as wonderful as it first sounds."

I was pondering about this point, then decided that I wasn't really eager to admire their equipment. Instead, I asked: "What about programming languages?"

"That is a huge department of its own. We use thousands of

languages, and some of them are so sophisticated that no amount of paper woeld suffice to hold a complete listing, so they are permanently kept on Womm, and you enquire only about what you need at the moment".

"What is Womm?" I asked, now suddenly being aware that it was me who was behind. But Flagbit didn't scorn my ignorance, or at least he concealed it magnificently and replied:

"That is our new word organising mass memory device. It is the first of its kind having an infinite capacity. Its access speed and transfer rates are still slow, but they are working on it. It has revolutionised our entire business and opened the door to a new generation of programming languages".

"I bet. But, I beg your pardon for asking, what are the goals in designing all these languages? After all, languages were invented to raise the quality, reliability, efficiency of systems, and to reduce the cost of their production", I suggested cautiously.

"Now, come on, Cleverbyte! That sounds pretty old-fashioned, even by earthly standards! I reckon you had a problem with unemployment lately too; up here it is one of major proportions. To be quite frank, it is directly responsible for the software explosion. Producing languages to make programming easier and simpler would be counterproductive. On the contrary, these languages are ideally suited to keep uncounted people on their intellectual toes, content and busy, and to maintain an image of progress and sophistication. We have whole armies of clerks writing manuals; and they love it".

I wasn't quite prepared for a sermon of such length, and it took me some time to digest this philosophy. So I asked naively:

"But have you discovered a way to comprehend these languages and profit by their use?"

"One never understands the whole thing. It is another of those stifling high-brow dogmas that one should be able to understand The Whole. When yoe are to solve a problem, you study the relevant sections of your language, and if you can't follow it, you take a course or have somebody write another manual for you. There are lots of souls waiting for attractive suggestions to teach a course or write a manual. Naturally, this will take too long if you have a genuine desire to get some problem really solved. Then you go back to first principles and simple means - just look at my cards! But it takes people a long time to find this out, just as with loosing their illusions about wishing".

"And how do you think this will be in the future? More mountains of manuals?"

"We don't really worry about the future, but if you care to know, just make a wish to be transferred to some language design committee", Flagbit remarked. We agreed that this was the best way to obtain a representative picture, for Flagbit had assured me that in the future all this was going to be done by committee. There followed a slight tremor, we were whisked away and found ourselves in the midst of a select group of obvious experts in full action. The scenery was splendid, a phantastic combination of seaside and mountain resort, making it particularly difficult for me to follow the subsequent discussion.

W: "The problem is one of coercions rather than types".
H: "Coercions can give one an amount of uniform reference which is beneficial".
I: "But the semantics change with uniform reference, that is, one has punning".
G: "Visibility is important and it must be taught as a practical concept".
I: "Visibility is conceptually hard".
D: "Visibility is tough, because of its interaction with block structure".
I: "Let us now discuss partial visibility!"
L: "A variable is like a capability".
G: "To beliebe that every variable is a reference is inaccurate. If we have sorted out visibility, then partial visibility will be easy".
R: "Algol 68 has the notion of possessing and referencing".
K: "A name cannot possess a reference!"
G: "All visibility should be coupled to compilation units".

I soon got restless for I could hardly perceive that they were talking about our subject at all. I was just about to voice my complaints when all of a sudden the whole region fairly rocked under the crash of four thousand and ninetysix thunder blasts.

"There, that's the Professor!" Flaggy whispered.

"Then let's be moving along", I urged, being anxious to leave this committee where I felt uncomfortably incompetent.

"Keep your seat, Cleverbyte", Flagbit said, "he is only just telegraphed".

"How?!"

"These blasts only mean that he has been sighted by our

computerised early warning radar system. He is just off Cape Canaveral. The committee will now go down to meet him and escort him in. But he is still millions of miles away, so the show won't come up for a considerable time, yet".

We walked down to the Conference Center at leasure. I was truly amazed by its sheer size; onehundredandthirtyonethousand and seventy two seats, virtually all empty.

"This venue looks pretty deserted. I bet there is a hitch somewhere again", I remarked pessimistically. "Is he perhaps going to give one of those highly specialised talkc that only a handful of experts can follow?"

"Don't you fret, Cleverbyte, it's all right, you shall see. Of course he is going to talk specialised. That is important to maintain the proper image. But the topic is just for show anyway. People will come for social and commercial reasons; it is fashionable to have been here, and you meet friends".

At his moment there was another big bang, like that of a new supernova.

"The Professor is through the security checks now and will be on stage within seconds", Flagbit explained. And then there was a big flash, the whole place was splendidly lit up, and all the seats were suddenly taken. My chin dropped a few inches by surprise, and my friend commented with undisguised pride:

"That's the way we do it up here. Nobody worrying about being late, nobody sneaking in after the curtains went up. Wishing is quicker than walking!"

However, I spotted a slight disturbance not too far from us. Somebody was making a distinct fuss.

"What's going on down there, Flaggy?" I asked, finding this scene somewaht unusual in such a well-organised place.

"You see, since all people wish to be in the stadium when they hear the bang, computation of seat assignments presents a few problems", he remarked with calculated understatement and his pride had visibly diminished. Then he continued:

"They (his previous We now had become a deferential They!) have recently put a new supercomputer into operation, but occasionally there is still a glitch in the algorithm, although it was announced to be formally verified. It doesn't take long until you realise that every verification is worthless as long

as it is itself not verified. And now this real-time algorithm is particularly sophisticated. It made great headlines under the name 'Seat assignment of the fly'".

After the excellent talk on "Quaternionic complexity on a three-tape Turing-machine without pointers" we had a drink and relaxed in the lobby. Our glasses not yet empty, a hefty, square-jawed man approached our table. Flagbit jumped up and pulled me by my sleeve.

"This man is the chief brain of our supercomputer. Hello, Megachip, let me introduce you to my friend Cleverbyte who has just arrived! What are the latest figures on your machine?"

"Well, it now works with three times the speed of light. Sixteen billion active elements placed on 2.56 million single chips!" was his reply.

By that time, I had already learned to keep my composure when hearing of staggering innovations, but nevertheless I must have been looking pretty foolish, for Flagbit interjected:

"You must know, Cleverbyte, Megachip has had the greatest idea ever: by making chips work faster than light you can read out your computed results virtually before you insert your data, provided you position your output station at a location remote from your input device".

I was at a loss for words and could merely state the obvious: "But this must revolutionize the entire computing business, particularly programming". Megachip laughed heartily:

"It sure does! All this craze about optimization is over. We have a store of several gigabytes, hundredtwentyeight thousand parallel microprocessors, sixteen thousand data channels running at megabaud rates. The whole hardware merely costs eight million pounds, which I am sure is not more than a handful of shillings was in your earthly days".

"This is a staggering feat indeed; but does your software stand up to these measures? I am sure its cost was immeasurably larger", I commented.

"The biggest single piece of software ever developed! The operating system alone takes over one billion bytes of instructions, and together with the compilers it took seventeen hundred man-centuries to develop, in spite of our loss of interest in optimization. Most of the work went into maintenance, and after several breakthroughs in reliability we

now have only about 50 breakdowns per second. The real turning-point was the acceptance of the fact that a perfect, faultless system would never materialise, but that instead we had to work towards a fault-tolerant, self-recovering system. This resulted in close to 100% of the breakdowns being recovered by the system without intervention".

"These are truly awesome figures to me! But may I ask you, Mr. Megachip, how this tremendous system was developed, and in particular what programming language you use, for with the ones that I know such a feat would have been utterly impossible".

"Well, Cleverbyte, you are right, but not quite. The language used has gradually evolved by extensions in all possible directions. To give you a measure, the manuals measure threehundred thousand volumes and the compiler uses up half a billion bytes. You see, we are quite willing to let anyone keep his habits of programming; hence the language must be compatible with pretty well every previous language that has ever existed".

This decidedly began to amuse me, for I felt I had heard these arguments before. Tongue in cheek I asked:

"But doesn't this inflation of languages largely offset the gains you have made in the development of hardware? I am convinced that this diversity and the systematic retention of old mistakes is a wasteful deadweight for men and machines alike".

"Of course it is; but listen, dear Cleverbyte, we sorely need this deadweight! You just take the wrong viewpoint; this bulky software does <u>not offset</u> our hardware innovations, but <u>justifies</u> them. By Jove, how could we otherwise find any motivation in our continued hard labor? Just think of it, we are in heaven where time is eternity and speed doesn't truly matter!"