OUTLINE OF PROPOSED WORK - C.A.R. Hoare

The most pressing problem in computing in the next twenty years will be the reduction in the cost of programming to match the reductions in hardware cost, some of which are already achieved and some are yet to come. It is this problem which I wish to address myself to in a period of full-time study. I believe that my experience and background, together with the help of colleagues at The Queen's University and the assistance which I can obtain from computer scientists throughout the world, give me some chance of showing a way in which this problem can be tackled.

The problem is a sufficiently serious one that must be approached on several fronts:

1. Improved education, in which the academic ideals of elegance, rigour, and accuracy can be brought to bear on the production of more efficient, more adaptable, and more reliable programs.

2. Improved tools; the tools of the programmer are his languages, his operating systems, his libraries and applications packages, his data base management systems, etc. Some of these tools are at present complicated and unreliable, inefficient, unstable, expensive, and too often ineffective.

3. Improved hardware: when the correct approach to the other problems has been outlined, the solutions can be further aided by the design of less inappropriate hardware.

I put forward the hypothesis that the eventual solution to these problems is orders of magnitude simpler than most people believe. Without such a hypothesis, it would be foolish for one man to embark on a project that has proved too much for teams of a thousand. I wish to test this hypothesis by showing how a very wide range of computer uses and applications can be built reliably and efficiently upon a very small basis of logical methods and reasoning.

My approach will be similar to that adopted by early workers in the Foundations of Mathematics, who showed that all mathematical deduction could be

carried out with the simple tools of the propositional and predicate calculus, and that the postulates of existing branches of mathematics could, by a suitable constructive definition of the primitive terms, be modelled as theorems of set theory. Similarly, I hope to show that the various "branches" of programming (eg commercial data processing, data base management, real time programming, symbol manipulation, and even engineering calculations) can be "reduced" by appropriate constructions (ie. library programs and packages) to a rather small number of universal programming concepts, systematically and simply applied. Such a demonstration would take the form of a textbook which would serve simultaneously as an aid to the education of computer scientists, and a detailed manual for users of the software described. Eventually, with the aid of many other contributors, and with the passage of time, such a text might become a standard, like texts in numerical programming or Knuth's work on algorithms, or Kleene's Mathematics; and the role of suppliers of general purpose software will be reduced to that of providing faithful machine-readable copies of the software described. In the further future, even hardware designers may be persuaded to provide the hardware on which that software will run most efficiently and reliably.

Of course, the analogy with the foundations of mathematics must not be pressed too far; some of the fascinating and deep foundationary problems of mathematics are not directly relevant for computer programming, which has several problems of its own.

(1) The "inefficiency" of mathematical and logical constructions is not a matter of concern; but computer programs must at least be reasonably efficient, or they will not be used, however elegant, reliable or powerful they may be.

(2) It is well known that all computer programming can be done on a Turing machine, which has the desired properties of simplicity and logical rigour. But to take these as the logical basis of all computer programming would be neither pleasant nor profitable.

I hope to show that both these dangers can be avoided; if I fail, I hope that an elucidation of the reasons for the failure will in itself be illuminating. For this reason, I regard the risk as worth taking.