Computer science education—The need for interaction

by M. STUART LYNN

Rice University
Houston, Texas

I would suggest that one of the central problems of computer science today is understanding what it's for. As a consequence, given this hypothesis, from an educational standpoint it is not clear what students are expected to represent when they graduate with degrees in computer science. To the potential employer, the situation becomes even more confused when (in better times) he believes he is recruiting one thing and finds that he has hired something else.

This lack of understanding and eventual misrepresentation is not surprising in a young discipline. Perhaps one of the difficulties faced by computer science is what may be an over-eager attempt to formalize its content and boundaries, and in so doing discouraging the growth, change and vitality which have characterized the last decade. Perhaps, also, there is an increasing role being played by what may be (but should not be) an underlying conflict between the systematic and formalizing demands of science, and the engineering demands by which concepts are made useful to man; this conflict underlies much recent uncertainty in education in other disciplines, and it should not be surprising that it is becoming increasingly present where computing is concerned.

An example, in the Curriculum 68 Report,^ of where potentially harmful boundaries exist lies in the treatment of numerical analysis. On the one hand, insufficient priority is given to this area, insofar as the proposed course contents are only suggestive of the field and give no insight as to the relationship of numerical analysis to the domain of computer science or to the domain of uses of numerical analysis in computer applications. On the other hand, excessive priority is given to numerical analysis when compared with other uses of mathematical analysis in computer applications: statistics, signal analysis, pattern recognition, control theory, mathematical programming, operations research and others. The proper role of each of these and related areas, and of their interrelationships, needs to be determined by examining the ways in which they are put to use in computer applications, with what priority, and then working backwards to define the appropriate educational structure. It is not clear that the existence of numerical analysis in the computer science curriculum should be an end in itself.

Perhaps too much of computer science has been concerned with formalization. To a degree, such concern is proper; there is clearly a need to develop and understand, systematically, fundamental principles and the interaction of these principles with each other; and a further need to educate in such a way as to encourage growth based upon a formalized body of knowledge. This should not, however, be the end in itself, as appears to be the trend.

This trend, if it exists, is not directly the fault of the Curriculum 68 Report. It occurs partly perhaps because Curriculum 68 is too comprehensive, or too ambitious; and since in practice limited budgets may dictate that only selected parts are implemented, an unbalanced educational structure may be achieved. Perhaps, too, Curriculum 68 is guilty of leaving to chance (see p. 155) what may be the most difficult question of all; how to effect the interaction with other disciplines which is so essential to the vital development of computer science. Amarel^ has separated computer science activity into its synthetic and analytic components, the former having a "pragmatic flavor largely responsible for major advances in computers and for great diversity of areas in which computers are being applied", and the latter "... providing conceptual guidelines". He advocates a continuous interaction between these two components as essential to a vigorous rate of progress in the field. I would suggest that education in computer science, as structured in Curriculum 68, may be guilty of inhibiting that interaction, not encouraging it.

More positive steps need to be taken to encourage this interaction between the science of computing and the uses of computers. I am not suggesting interaction does not exist, only that it has received decreasing emphasis in recent years. Perhaps such encouragement
cannot be achieved solely within the definition of computer science education itself, but requires closer examination of the administrative structures within which computer science operates.

It is likely that the industrial employer will become increasingly selective in its recruiting at all levels. This will be dictated not only by short-term considerations, such as the current state of the economy, but also because he will primarily be looking for graduates who have the breadth to be able to understand and deal with a wide spectrum of his problems, graduates who have the flexibility to modify their areas of interest as needs and priorities change. To the extent that education in computer science prepares students for their future, and for a considerable number this may mean industrial employment, Curriculum 68 needs careful reexamination, since it is not clear that it contributes substantially to this requirement.

A mathematician has been defined as one who, given a problem to solve, solves a whole class of problems of which the original is not a member. It can be argued that there has been a breakdown in the lengthy communication channels between mathematics and the environment in which it operates: whether this is good or bad for mathematics is a matter of point of view. It would, however, be bad for computer science. If students are to be prepared for what may be ahead, they must be given perspective, and a realistic insight into how computer science fits into and relates to the world around it. They should not, to reverse Oscar Wilde's definition of a cynic, be encouraged to know the value of everything and the price of nothing.

REFERENCES

1. Curriculum 68
   Comm of the ACM 11 1968 p 151
2. R W ELLIOTT
   Master's level computer science curricula
   Comm of the ACM 11 1968 p 507
3. S AMAREL
   Computer Science: a conceptual framework for curriculum planning
   Comm of the ACM 14 1971 p 301