INDUSTRY REACTION TO

COMPUTER SCIENCE EDUCATION

Position Statements by
Panelists

Dr. George Dodd
Computer Technology Department
GM Research Labs
Warren, Michigan

Today, undergraduate Computer Science Education teaches mechanics without teaching problem solving. Typical curricula include courses in assembler languages, compiler theory, list processing, and automata theory. Every MS degree holder, and most BS degree holders, know Polish notation and have written parts of compilers. However, few of them have ever learned to write a program that can be easily enhanced or respond to changes as new management (instructor) requirements are set forth. Even fewer can read a program and describe what it does or debug a system consisting of ten or more modules.

I propose that undergraduate computer science education be refocused to teach problem solving. The teaching of programming and other mechanical skills should be relegated to laboratory discussion sessions. Nonnumeric computer problems should be emphasized more (how many business computer professionals ever invert a matrix?). Engineering and business options should be established to give students special emphasis in these disciplines.

Graduate level computer scientists should be trained for education, for research and development and for management. Management options should include courses in accounting and business and liability law. Computer professionals at all levels should learn how to design computer systems capable of interacting ina reasonable fashion with the non-computer oriented human world.

Fred A. Gluckson
EDP Systems Department
National Bank of Detroit
Detroit, Michigan

Industry and the university must operate in a close partnership, since neither can exist very long without the other. But the partners seem to take each other for granted and, for the most part, do not communicate effectively. There needs to be more exchange of ideas and of people in order for industry and universities to better serve each other.

My belief is that there are too many <u>computer science</u> programs at the university level and <u>too few information processing</u> curricula. The four-year institutions, with some exceptions, are teaching what they know how to teach rather than what they should teach. Consider a typical course description: Numerical Analysis II covers "matrix operations; evaluation of determinants; solution of systems of linear equations and matrix inversion; calculation of eigenvalues and eigenvectors."

Courses such as this may be counter-productive. They give the student unrealistic expectations of industry, and discourage college recruiters. To quote Dr. Richard Hamming of Bell Labs "The universities aren't turning out a product that the country wants. I hear people in industry say they will never again hire a computer science major".

Business data processing is characterized by the manipulation of large inter-related data files. Many graduates never had the opportunity to work with these. Universities should have a more practical approach to computer science education. If necessary, people from industry should consult to or even teach such courses.

Fred S. Patterson
Computer Communication Services
Dow Corning Corporation
Midland, Michigan

It is my contention that the Computer Science curricula of today is doing an adequate job of presenting the theory and the practice of Computer Science. This can be seen in the numerical analysis and systems areas as well as in the artificial intelligence area. However, as far as industry is concerned, it is not the in-depth knowledge of Computer Science field, particularly, systems development, but rather, a breadth of knowledge going across many differnet disciplines including, the technologies, as well as communication sciences. If these broad capabilities are not apparent, it would be almost impossible to communicate with the functions for which the systems are going to be developed.

Possible solutions to this problem or ways of making the transition from the university to industry easier have been suggested. One of these is the possibility of a Co-op Program whereby university graduates or those on the verge of graduation, perhaps between their junior and senior year would actually spend some time in industry developing systems with trained systems analysts. They would be gaining both experience on the job, as well as a better understanding of the kind of things that their education is lacking. Another suggestion is the possibility of the university instructor actually spending some time with the industry in learning the kind of things that his people would be expented to know. Still another way might be to allow people in industry to actually teach courses at the university, bringing in the practical aspects of systems development. Any of these proposals would certainly benefit the new graduate in gaining the ground needed to excel in the world or Computer Science.

Prof. Thomas J. Schriber
Graduate School of Business Administration
University of Michigan
Ann Arbor, Michigan

One of the more subtle problems in computer education is that of preparing Business School students for intelligent use of the computer resource in their eventual career. The problem has several dimensions, some of which can be appreciated by considering a series of questions. What should the future manager know about algorithmic processes and programming languages? How much programming, if any, should be required as part of the future manager's education?

What should the future manager know about such data processing topics as number systems, information retreival, sequential and random-access files, input-output devices, telecommunications, management of the data processing department, the economics of computer use, and the probable future impact of technological change on the data processing area? How can the computer be integrated meaningfully into such non-computer courses as accounting, finance, marketing, production and operations management, and business policy? Or can this be done at all? To what extent should pre-programmed routines be used by the student? How can the student be made comfortable vis-a-vis such routines, using them intelligently and questioning them in critical but constructive fashion, instead of using them blindly, as black boxes? How can the faculty who traditionally teach in non-computer areas such as accounting, finance, etc., best prepare themselves to expose their students to up-to-date use of the computer in these areas?

In the educational process, what is the tradeoff between having the student spend his time non-creatively in the mechanics of computer use (for example, traveling to and from the computing center, waiting in line at a teletypewriter or a card punch, punching cards, waiting for output), vs. spending time in the library studying the conceptual aspects of the subject proper (e.g., accounting, finance, etc.)? There is no consensus on answers to questions such as these. The panel will explore some of the pertinent issues involved, and invite critical commentary from the audience.