A Comparison of the ACM/C's and the IEEE/CSE Model Curriculum Subcommittee Recommendations

Gerald L. Engel
Virginia Institute of Marine Science

Introduction

The year 1977 saw the independent development of two significant documents in the field of computer science and engineering education: "Curriculum Recommendations for the Undergraduate Program in Computer Science: A Report of the ACM Committee on Curriculum in Computer Science,"1 and A Curriculum in Computer Science and Engineering2 prepared by the Model Curricula Subcommittee of the IEEE Computer Society. The work leading to the publication of these reports was independent, though the committees did maintain liaison.

The work leading to the report of the ACM Curriculum Committee on Computer Science (C'S) is an outgrowth of the earlier reports of that committee—most significantly the 1968 effort known as "Curriculum '68."

Both the ACM and the IEEE Computer Society documents represent many man-years of effort. Significantly, in both cases the second general set of recommendations followed by approximately 10 years the appearance of the first major report.

Predictably, the reports differ. It would appear that there are still differences in background and philosophy between the computer engineer and the computer scientist trained in the liberal arts tradition. What is more important, however, is the significant degree of overlap between the reports in spite of the differences in orientation of the program designers and the independent development of the two sets of recommendations. We will investigate the extent of overlap in more detail below; however, the similarities that so clearly exist are one measure of the maturity of the field: differing groups can clearly define a common core of information required of all who wish to enter the field.

Background of the reports

The work on the C'S report began under the chairmanship of Preston Hammer in 1974 with a meeting at the Virginia Institute of Marine Science. Details of this meeting appear in a report by Barnes and Engel3 and the working papers from the meeting appeared in the September 1974 and December 1974 issues of SIGCSE Bulletin. Among the early results of this work was the publication of a comprehensive bibliography of the literature of computer science education since "Curriculum '68,"4 which was used as a basis for further work.

Following the 1974 meeting, public panels and sessions were held to consider the report that was in development. In this manner the entire membership of the ACM was in a position to have input to the report. Through this process (or in spite of it) the report was completed in February 1977. It is this version that appears in the June 1977 SIGCSE Bulletin. This report is itself regarded as a working document and is being circulated in this manner to receive as much input as possible prior to the preparation of final recommendations. The same issue of SIGCSE Bulletin reported on another important curriculum development effort dealing with programs at the two-year level.

The development of the IEEE/Computer Society recommendations came from the work of a somewhat more structured and restricted committee, although preliminary position papers were published in Computer, and three panel presentations of the work were given. The effort required approximately 2½ years and resulted in the publication of a review draft, which was circulated among a number of interested professionals. The results of the evaluation were factored into the final report; in fact, the entire evaluation appears in the curriculum document.

Since the recommendations were published, the Computer Society has been helping institutions develop plans for implementing the program. This effort, known as the Regional HELP Program, took the form of three workshops—one in June 1976 at Illinois State University, another in March 1977 in San Francisco, and the third one in June 1977 in Williamsburg.

In summary then, the reports developed on different lines, with the C'S activity bringing more active involvement of the ACM membership throughout the design phase, while the Computer Society focused its primary review emphasis during the completion phase. Both projects were funded through the professional societies and conducted through the efforts of volunteers.
The ACM recommendations

Key to the CS recommendations is an eight-course sequence of core materials:

CS-1. Computer Programming I
CS-2. Computer Programming II
CS-3. Assembly Language Programming
CS-4. Introduction to Computer Organization
CS-5. Introduction to File Processing
CS-6. Operating Systems and Computer Architecture I
CS-7. Data Structures and Algorithm Analysis
CS-8. Organization of Programming Languages

These are assumed to be 3 semester-hour courses; their prerequisite structure is shown in the Figure.

ACM’s prerequisite structure. The solid lines indicates definite prerequisites; the dashed lines are recommended prerequisites.

The committee recognized that other implementations may be desirable. At the elementary level, the report specifies a series of topics that represents one of many possible implementation structures.

The report goes on to suggest additional elective courses in computer science at the intermediate and advanced levels.

CS-10. Computers and Society
CS-11. Advanced Systems Programming
CS-12. Minicomputer Laboratory
CS-13. Database Management System Design
CS-14. Analysis of Algorithms
CS-15. Theory of Programming Languages
CS-16. Compiler Writing Laboratory
CS-17. Automata, Computability, and Formal Languages
CS-18. Numerical Mathematics: Analysis

The dynamics of the field also lead to the recommendation of special topics courses where departmental resources are sufficient. As these “topics” mature one would expect that some of the material would be integrated into courses previously specified—or indeed might replace entire courses in the curriculum. Included in this area are the following:

A. Microprocessor Laboratory
B. Performance Evaluation
C. Telecommunications/Networks/Distributed Systems
D. System Simulation
E. Graphics
F. Artificial Intelligence
G. Structured Programming
H. Software Engineering: Large Systems Design
I. Topics in Automata Theory
J. Topics in Computability
K. Topics in Formal Language Theory
L. Simulation and Modeling

Finally, tied to the recommendations is a structure of mathematics. However, it should be recognized that such material is not prerequisite to the core material of the curriculum. Included here are the following:

MA1. Introductory Calculus
MA2. Linear Algebra
MA3. Mathematical Analysis I
MA4. Discrete Structures
MA5. Mathematical Analysis II
MA6. Probability and Statistics

The undergraduate program, then, consists of the eight courses of core material plus four additional courses selected from the recommended computer science intermediate and advanced electives with no more than two in any one specific subfield of the discipline. As the student progresses through the computer science portion of the program he begins at a very practical level; as he moves ahead the work becomes more conceptual and theoretical.

It is further recommended that the computer science major take mathematics courses MA1, MA2, MA3, and MA4, and—depending on the electives selected—MA5 and MA6. This represents a minimum requirement for the major in computer science at 48 semester hours, which is less than one-half the typical undergraduate program.

The details of the curriculum are, of course, contained in the full report, and the interested reader is referred to that document.

The Computer Society recommendations

The Computer Society’s recommendations are discussed in great detail in other papers in this issue and thus will only be briefly sketched here. The Core Curriculum of the recommendations encompasses portions or all of the following:

1. Introduction to Computing (with laboratory)
2. Introduction to Computer Organization
3. Data Structures I
4. Mini-computer Laboratory
5. Microprocessing Systems (with laboratory)
6. Digital Logic I (with laboratory)
7. Digital Logic II
8. Programming Languages
9. Discrete Structures
10. Design and Analysis of Algorithms

The report refers to course modules. Modules 1-5 are, in their entirety, required in the core, while modules 6-12 are required in part, except that the digital laboratory associated with module 6 is fully required. As with the ACM report, the partial classification of modules indicates that core areas are really specified by a series of topics that can be implemented into courses in a variety of ways to meet local circumsstances.

The core curriculum is then supplemented by a variety of courses that are placed in the broad areas of digital logic, computer organization and architecture, software engineering, and theory of computing, to fill out the major program. Within the course structure, only Digital Logic I and its associated laboratory, and Introduction to Computing and its associated laboratory are considered...
as “introductory.” The remainder of the material is listed as “intermediate.” “Advanced” material, at least by implication, is left to the graduate curriculum. The credit-hour requirements of the core are roughly equivalent to those of the ACM Curriculum.

In addition to the material listed thus far, additional course modules are discussed in the following areas:

- Digital Logic Devices (and laboratory)
- Digital Design Automation
- Microprogramming
- Computer Architecture
- System Design Laboratories
- Distributed Processing and Networks
- Operating Systems and Computer Architecture II
- Data Structures II
- Data Base Systems
- Translators and Translator Writing Systems
- Automata and Formal Languages
- Theory of Computing

Mathematics requirements in and of themselves are not stated within the Model Curricula, nor is any indication given of course work in the area of societal impact.

A comparison of the reports

As can be seen from the discussion thus far, there is a strong overlap between the two reports. The overlap is, in fact, so great that both groups engaged in a simultaneous though independent discussion of whether or not the areas of architecture and operating systems should be combined in a single integrated presentation.

At the risk of slight oversimplification, the curriculum structure may be presented as three general elements:

| HARDWARE AND LOGIC DESIGN | SOFTWARE ENGINEERING AND PROGRAM DESIGN | THEORY |

In the philosophies expressed in the two curricula documents, the design of the ACM Curriculum relies more heavily on the right side of the diagram, while that of the Model Curriculum Subcommittee places more emphasis on the left side. The central portion, which encompasses virtually all of the core curricula of the two reports, and includes virtually all of the areas of software engineering and program design together with portions of hardware and logic designs and theory, are common to both sets of recommendations. Considering that the two pieces of work were developed independently, this is indeed remarkable.

Conclusions

The two documents show remarkable similarity (in software engineering and program design) while still showing a degree of diversity (in the areas of hardware and logic design, and theory). Despite the diversity, each group recognizes a significant level of importance in the basic areas of the other group, as demonstrated by each group’s definition of the core requirements.

The fact that the C'S and the Model Curriculum Subcommittee were in close agreement regarding a common core of material suggests that the future work of these groups might be profitably carried out in closer cooperation. One day, perhaps, computer science and computer engineering will no longer exist as separate entities, but instead as a single program representing options of a common core of fundamental material. This common core is the essence of our profession.

References


Gerald L. Engel is head of the Department of Computing and Statistics at the Virginia Institute of Marine Science, Gloucester Point, Virginia. Prior to this, he taught at Randolph-Macon College, Hampden-Sydney College, and Pennsylvania State University. His work in computer education has included active participation on the ACM Curriculum Committee on Computer Science, the steering committees for the conferences on computers in undergraduate curricula, and the NSF sponsored study of computers and society courses for ACM. He currently serves as liaison between ACM's Education Board and the Education Committee of the IEEE Computer Society.

Engel received the BS degree in 1964 from Hampden Sydney College, the MA degree in 1965 from Louisiana State University, and the DEd degree in 1974 from Pennsylvania State University.

December 1977

123