Curriculum 2001: Bringing the Future to the Classroom

Carl Chang, Gerald Engel, Willis King, Eric Roberts, Russ Shackelford, Robert H. Sloan, Pradip K. Srimani

The discipline of computing encompasses the understanding, design, and use of computers and computational processes. The breadth of the discipline is emphasized in the following quotation from a report issued by the Computing Sciences Accreditation Board. [1]

> The discipline ranges from theoretical studies of algorithms and computability to practical problems of implementations in terms of computational hardware and software. Thus, the discipline spans both advancing the fundamental understanding of algorithms and information processes in general as well as the practical design of efficient reliable software and hardware to meet given specification[s] ... . [I]t includes theoretical studies, experimental methods and engineering design in all disciplines.Computing draws on the methodologies of both science and engineering. Theoretical work has done much to advance the state of the art. At the same time, computing does not separate the discovery of new scientific knowledge from the application of that knowledge to solve practical problems. The intimate relationship between theory and practice endows the discipline with much of its strength and vitality. This same connection between theory and practice, however, also means that the body of knowledge associated with computing changes very quickly as technology evolves.

Keeping Pace with Technology

The rapid evolution of the discipline presents a special challenge for professional societies such as the IEEE Computer Society and the Association for Computing Machinery. To best serve educators in computer science and engineering, these organizations have continually provided updated curricular guidelines in response to these changes.

Efforts to design model curricula began in the 1960s. In 1968, building on a series of earlier studies,[2,3] the ACM published *Curriculum 68,*[4] which offered a detailed set of recommendations for academic programs in computer science. In 1977, the Computer Society published *A Model Curriculum for Computer Science and Engineering,*[5] which was the first report to bridge the gap between software- and hardware-oriented programs. The ACM revised its curriculum in 1978,[6] while the Computer Society updated its computer science and engineering curricula in 1983. [7] In the late 1980s, the Computer Society and the ACM joined forces to undertake a more ambitious curriculum review, which was eventually published as *Computing Curricula 1991.* [8]

In the fall of 1998, the Computer Society and the ACM appointed a joint task force on "Year 2001 Model Curricula for Computing: CC-2001," with a charter to review the Joint ACM and IEEE/CS Computing Curricula 1991 and develop a revised and enhanced version for the year 2001 that will match the latest developments of computing technologies in the past decade and endure through the next decade.

Core Area Guidelines and Model Curricula

From the time of the first report, the curriculum projects have provided guidelines for academic programs in computing, pooling the expertise of leaders in the field to define what undergraduate students in computer-related majors should learn. The format of the reports has varied over the years. The 1978 ACM report [6] provided detailed course descriptions for curriculum makers to follow. The 1983 Computer Society report [7] recommended laboratory materials to support lecture topics and used a modular approach to organizing subject matter.

The 1991 report, the first joint effort of the two societies, built on the 1983 efforts and introduced collections of subject matter modules called *knowledge units* that comprise the core curriculum requirements for all undergraduates. This report also included guidelines for advanced courses.

Assessing the Impact of Curriculum 1991

The first major initiative for the Curriculum 2001 task force was to assess the impact of *Computing Curricula 1991.* To accomplish this, the task force designed a simple questionnaire (see the sidebar "Survey Questions for *Computing Curricula 1991*"), which was mailed to the chairs of computer science departments worldwide and also made available through the Web.

We received 124 responses through the Web and about 30 responses through regular mail. Of these, more than 98 percent supported the idea of updating the curriculum guidelines. And although most respondents liked the concept of defining knowledge units, many strongly supported a more concrete definition of a minimal core required for all computing students—although opinions varied as to what that core should contain. In addition, many respondents wanted to see a greater emphasis on course design. The respondents also expressed interest in having the final report

define more model curricula, taking into account the diverse educational resources and requirements at academic departments throughout the world. The survey results strongly suggest that the revised curricula should pay greater attention to accreditation criteria for computer science and engineering programs.

Principles

The Curriculum 2001 task force will build on the work done by past committees—particularly the overall definition of the discipline and the philosophies of computing education they express. From our study of the Computer Society's and the ACM's past reports and our survey of the impact of *Computing Curricula 1991*, the task force has adopted ten principles.

Future computing curricula should have dual objectives: to produce both excellent researchers and excellent practitioners.

Curriculum 2001 should produce college graduates who can excel in graduate programs, becoming the kinds of computer scientists and engineers who will open new doors through research. At the same time, the curricula outlined by the task force should give students problem-solving skills they can apply to real-world problems.

Computing integrates mathematics, science, and engineering.

We strongly endorse the position articulated in *Curriculum 1991* that "mastery of the discipline includes not only an understanding of [the] basic subject matter, but also an understanding of the applicability of the concepts to real-world problems."[8] Particular attention must be paid to the importance of laboratory work, which helps students apply what they have learned to diverse scenarios.

Knowledge units are valuable in the process of curriculum design.

Individual knowledge units, the key structural component for *Curriculum 1991*, provide the framework for the design of individual courses as well as entire curricula. We will update the set of knowledge units, incorporating new developments in computing from the past 10 years. By identifying a flexible set of units, the new curriculum guidelines can accommodate the needs of a broad range of institutions.

Curriculum 2001 must offer guidance in individual course design.

Although knowledge units represent valuable tools for curriculum design, many institutions need prescriptive guidance at a more detailed level. Our survey of the impact of *Curriculum 1991* revealed that many institutions continue to work with the models outlined in *Curriculum 1978*, including specific course designs. For such institutions, *Curriculum 2001* will be most effective if it defines a selection of model courses that combine the knowledge units into reasonable, easily implemented packages.

Curriculum 2001 must identify core concepts that should be presented to all students.

The increased presence of computers in society has historically led to a parallel expansion in the computer science core at universities: As important new topics emerge in the field, many programs attempt to include them as undergraduate requirements. Because of the constraints of an undergraduate degree, programs cannot include new topics like software engineering, human-computer interaction, networks, and graphics while retaining the traditional presentation of classic topics such as assembly language programming, compiler construction, and automata theory.

To handle this increase in topics, perhaps the best strategic approach is to *reduce* the size of the required core, allowing greater flexibility to include new topics and adapt a curriculum as changes occur. The *Curriculum 2001* task force has agreed that "the core will consist of those topics for which there is a broad consensus that the topic is essential to undergraduate degrees that include computer science, computer engineering, and other similarly named programs." [8] The core does not constitute a complete undergraduate curriculum in itself, but must be supplemented by courses that may vary by institution, by field of study, or even by individual student.

Curriculum 2001 must provide guidelines for courses beyond the required core.

In addition to specifying core knowledge areas, *Curriculum 2001* must provide guidelines for intermediate courses, which cover fundamental knowledge units outside the core, and advanced courses, which serve as technical electives in specialty areas. The separation of knowledge units into three levels will offer flexibility to curriculum builders to respond to local situations.

Curriculum 2001 must be international in scope. It is our intention that *Curriculum 2001* be useful for computing professional around the world. While specific course outlines such as those provided in *Curriculum 2001* may not be useful for all localities, the extensive list of knowledge units should provide a solid framework for curriculum development anywhere.

The development of Curriculum 2001 must involve significant industry participation.

Most students who graduate from undergraduate computing programs go on to get jobs in industry. To ensure that our graduates are properly prepared for the demands they will face in these positions, we believe it is essential to involve industrial colleagues in the design and development of curricular guidelines.

Curriculum 2001 must consider professional practice skills.

In addition to engineering topics and theoretical issues, computer science and engineering undergraduate programs must cover the myriad other skills required for success in the profession. These include management skills, written and oral communication abilities, and the ability to work as part of a team.

Curriculum 2001 (continued from page 71)

Curriculum 2001 must meet the needs of undergraduate programs

So that academic programs around the world can benefit from its guidelines, *Curriculum 2001* must

- satisfy the requirements of accreditation programs worldwide,
- be compatible with model curricula of related disciplines including information systems, electrical engineering, and mathematics,
- accommodate the various emphases and objectives of computing programs, and
- accommodate future advances in the computing discipline in a timely fashion.

Knowledge Area and Pedagogy Focus Groups

The Curriculum 2001 task force has identified key knowledge areas that should be represented in the final *Curriculum 2001* report. These include mathematics and science, discrete structures, algorithms and complexity, architecture, intelligent systems, information management, human computer interface, graphics and visualization, operating systems, programming fundamentals, programming languages and translation, software engineering, net-centric computing, computational science, and social, ethical, legal, and professional issues.

With these domains as a starting point, knowledge area focus groups will brainstorm to generate a list of knowledge units essential to computing curricula. These groups of five to seven experts in each area will

- review and define the scope of the focus area,
- finalize the associated list of knowledge units,
- comment on the three processes of theory, abstraction, and design described in *Curriculum 1991,*
- separate the knowledge units into two or three levels, comprising the core, the intermediate, and the advanced subject matter,
- based on the knowledge units, suggest model courses and corresponding lecture or lab hours, with specific course objectives and expected learning outcome,
- select corresponding mathematics and physical science domains, and
- highlight changes made to *Computing Curricula 1991.*

Pedagogy focus groups will consider the knowledge units suggested by the knowledge area groups to consider the feasibility of studying these areas in computing curricula. The pedagogy groups will identify

- goals for the first year of study, critiquing the traditional programming-first approach and providing a short list of alternatives,
- supporting courses to complement undergraduate curricula,
- the computing core, or essential foundation, for computer science, especially seeking alternatives to the traditional curricular organization around computing artifacts,
- professional practices, establishing how this content can be integrated into the curriculum,
- advanced study areas, specifying the minimum number of courses required to round out the undergraduate experience and evaluating models for undergraduate research, and
- issues related to computing across the curriculum, or the "core of the core" material relevant to students in all disciplines.

Activities Of The Curriculum 2001 Task Force

With 30 years of history to build upon, new technologies and applications emerging every day, and a growing global concern for professional certification and accountability, the *Curriculum 2001* task force faces many challenges. First, we plan to expand and update the set of knowledge units encompassing the breadth of computer science topics now prevalent in industry. We will also identify the set of knowledge units that comprise the core of computing. Here, we will take a minimalist approach, including only topics that are deemed essential by a broad consensus.

Next, the task force will draft detailed descriptions of course sets that cover the essential core knowledge units. These descriptions will be specific enough to serve as a framework for educators, publishers, and curriculum resource developers. At the same time, the descriptions will be flexible enough to allow for institutional variations and effective evolution.

With these descriptions established, we will list and categorize additional courses that cover knowledge units beyond the core. These courses will form the basis for upper-division curricula.

When all of the elements are defined, we will prepare a strawman version of *Curriculum 2001* and make it available for public comment. After several iterations and revisions, we will submit the final report for approval by the IEEE Computer Society and the ACM.

Curriculum 2001 will be most productive and useful if it is bolstered by the widest possible public consensus. To achieve that consensus, we encourage comments, criticism, suggestions, and recommendations on the various drafts from anyone interested in computer science education. To find out more about the task force's progress on *Curriculum 2001*, visit <http://computer.org/educate/>.

References

1. Computing Sciences Accreditation Board, Inc., *Annual Report,* Stamford, Conn., 1998.
2. COSINE Committee, Commission on Engineering Education of the National Academy of Engineering, *Computer Science in Electrical Engineering,* National Academy Press, Washington, DC, 1967.
3. President's Science Advisory Commission, *Computers in Higher Education,* U.S. Government Printing Office, Washington, D.C., 1967.

4. ACM Curriculum Committee on Computer Science, "Curriculum 68: Recommendations for the Undergraduate Program in Computer Science," *Comm. ACM*, Vol. 11, No. 3, Mar. 1968, pp. 151-197.
5. Education Committee of the IEEE Computer Society, *A Curriculum in Computer Science and Engineering*, IEEE Publication EHO119-8, 1977.
6. R. Austing et al., "Curriculum 78: Recommendations for the Undergraduate Program in Computer Science," *Comm. ACM*, Vol. 22, No. 3, Mar. 1979, pp. 147-166.
7. Educational Activities Board, *Model Program in Computer Science and Engineering*, IEEE CS Press, Los Alamitos, Calif., Order No. 932, 1983.
8. ACM/IEEE-CS Joint Curriculum Task Force, *Computing Curricula 1991*, ACM, Baltimore, Md., Order No. 201880, 1991; http://computer.org/educate/cc1991/.

Survey Questions for *Computing Curricula 1991*

The Curriculum 2001 task force designed the following questionnaire to gauge the impact of *Computing Curricula 1991*.

1. Did you use Computing Curricula 1991 in any way in the past?
2. If you are a college or university teacher, do you know if your department ever looked at or used Computing Curricula 1991?
3. If you answered yes to either question, how was it used, and what features of it were helpful?
4. Do you think there is a need to create Computing Curricula 2001? Why?
5. Computing Computing Curricula 1991 had 10 main content areas. Do you think any new area should be added? Any existing area delet
ed? Any existing area updated?
6. Do you believe Computing Curricula 2001 should provide guidelines about a minimal core? If so, what would that be?
7. Do you have any suggestion about the format? Curricula 1991 was designed in terms of knowledge units and then possible model curric
ula in terms of those knowledge units.
8. Have you any other comments/suggestions for updating Curricula 1991?

Members of the Computer Society/ACM Joint Curriculum Task Force

Computer Society Members

James Cross, vice president of the Computer Society Education Activities Board
Carl K. Chang (task force cochair)
Gerald Engel (task force cochair and editor)
Doris Carver
Robert H. Sloan (secretary)
Dick Eckhouse
Michel Israel
Willis King
Francis Lau
Pradip K. Srimani

ACM Members

Peter Denning, chair of the ACM Education Committee
Russ Shackelford (cochair)
Eric Roberts (task force cochair and editor)
Richard Austing
Fay Cover
Andrew McGettrick
Mike Schneider
Ursula Woltz