Spectrum of Engineering Sciences
Vol-1, Issue-2
http://thecjlr.online/index.php/1/issue/view/2

Possible Directions for Computer Science Curriculum Development in the Future

Shoaib Khan
Ms Scholar Department of Computer Science Riphah University Islamabad
Abbas Afridi
Ms Scholar Department of Computer Science Riphah University Islamabad

Abstract

In recent years, there has been increasing scrutiny and criticism over the traditional balance between research and education, as well as the public's funding for universities. Currently, numerous industrial and scientific sectors anticipate a continuous scarcity of highly skilled computer scientists and engineers. This is because contemporary society is progressively relying more on digital technologies. Compared to many other academic disciplines, the gap between the fundamental principles of computing and the cutting-edge advancements in research and practical applications is rather small in the field of computing. As a result, the curriculum for computer science and engineering (CS&E) must continuously adapt to include new and important advancements. Moreover, swift advancements in technology impact the delivery of education. Advancements in networking and graphics have enabled the creation and distribution of instructional resources that offer significant advantages to a wide audience.

Keywords- Possible Directions, Computer Science, Curriculum Development

Background

In the fifty years that have passed since the beginning of computer science and engineering, a significant portion of that time has been devoted to ongoing research and development. Since the foundation of computer science departments in the middle of the 1960s, there has been a large amount of emphasis paid to the challenge of educating students in the rapidly developing field of computer science. In 1968, the Association for Computing Machinery (ACM) published Curriculum 68, which was a comprehensive document that outlined the curriculum for computer science. After that, the research was revised and published in two papers, Curriculum 78 (ACM 1978) and Curricula 91 (ACM/IEEE 1991), which are normally published every ten years. In 1983, the Educational Activities Board of the IEEE Computer Society (IEEE-CS) developed an independent curriculum for the study of computer science and engineering. In later years, the Association for Computing Machinery (ACM) and the IEEE Computer Society (IEEE-CS) collaborated in order to publish the Curricula 91 study. Several organizations have published additional reports on a variety of subjects, such as the structure of introductory courses (Koffman et al 1984, 1985), accreditation standards (Mulder and Dalphin 1984; CSAB 1987), and secondary school curricula (ACM 1993; College Entrance Examination Board 1996). These reports cover a wide range of topics, including computer science in liberal arts colleges (Gibbs and Tucker 1986; Walker and Schneider 1996). For the past seven decades, the Association for Computing Machinery (ACM) Special Interest Group in Computer Science Education (SIGCSE) has been organizing an annual conference that has witnessed a consistent increase in the variety of attendees.

Additionally, other computing forums, such as Communications of the ACM, have demonstrated a major interest in the subject of computer science education. Denning [1981, 1984], Dijkstra [1989], and Abelson et al. [1995] have all made substantial contributions to this field. There are several advantages to being a member of the community of computer science educators because of their propensity to explore and use a variety of instructional strategies. In addition to the conventional model, which places an emphasis on finding solutions to problems and programming, there have been other models developed in order to improve the efficiency of educational programs. A particularly striking example of this variation can be seen in the early curriculum formulation. A course of study that is modeled after the apprenticeship paradigm [Proulx et al., 1996; Rasala et al., 1994]. A curriculum that places an emphasis on inclusivity as a primary concern According to Tucker et al. (1995a, 1995b), According to Van Loan et al. (1996), computer science Patt's description of computer architecture from 1996 is shown here. In the year 1985, Abelson and his colleagues were the ones who first presented the idea of functional programming. - instruments of communication The work that Stein did in 1996 is the source that is cited. The year 1993 marked the beginning of Roberts's introduction of the idea of library-based programming. Decker and Hirschfield (1995), Astrachan (1996), and Conner et al. (1995) are the sources that were utilized in this examination. The year 1996 marked the beginning of Rudich's introduction of the idea of object-oriented programming. In addition, the process of development and experimentation occurs at various levels throughout the curriculum.

For instance, the Software Engineering Institute at Carnegie Mellon University has developed an all-encompassing undergraduate program that is centered on software engineering [Ford 1990; Shaw 1990]. In addition, members of the teaching staff at a number of different universities have designed computer science courses that have the potential to serve as models for implementation in other academic institutional settings. CSTB's "Computing the Future" study from 1992 and the Association for Computing Machinery's SDCR Workshop from 1996 both consider education to be a distinct subject, stressing the significance of education as a strategic field in the field of computer science and engineering research.

Areas of Concern in Cs & E Education

Attempting to handle the breadth of the field of computer science and engineering education within the confines of a single discussion session is an endeavor that cannot be accomplished. In order to guarantee the proper administration and structure of the subject matter, our working group has reached a consensus decision to divide the following domains from one another:

(1) general curriculum issues,

(2) undergraduate education,

(3) graduate education,

(4) K-12 education, and

(5) coordination within the education community.

Each of these themes is discussed separately in the following subsections.

General Curriculum Issues

Tucker and Wegner (1996) assert that the computer science discipline is always evolving, resulting in significant demands on the computer science and engineering curriculum. Continuously evaluating educational approaches and the subjects they cover is essential in light of the advancement of new instruments, tactics, and paradigms. Universities face a distinct problem in keeping their curricula up to date due to the rapid pace of advancements in various fields. Due to the rapid advancement of various fields and technology, the CS&E curriculum inevitably becomes outdated.

The gradual decline in the relevance of the core curriculum, which is required by many institutions, is indicated by the lack of a significant number of important disciplines. Subjects such as problem solving, parallelism, networks, program design, software safety, and the societal impact of technology exemplify the object-oriented paradigm. An effective approach to tackle the insufficient attention given to these issues would involve revising the knowledge units in Curricula 91 [ACM/IEEE 1991] to incorporate contemporary core subjects and themes, while eliminating outdated concepts. The ACM/IEEE-CS curriculum reports have historically been published every ten years. Therefore, it is crucial to create a methodology that is very adaptable and allows for the integration of newly generated ideas and objects. A comparable approach should be adopted to consistently identify and eliminate obsolete content from the core curriculum. Universities must establish more efficient ways for transferring instructional materials due to the swift advancement of computer science.

The educational community lacks coherence and undergoes constant reinvention. The proliferation of almost similar CS1 texts can be attributed to the excessive creation of mediocre outcomes and a waste of effort. In order to achieve cost-efficient and widespread distribution of effective course materials, it is necessary to create new platforms. The SIGCSE Web page [ACM 1996b] and the NSF Computer Science Courseware [National Science Foundation 1996] repository give links to a wide range of course materials from different institutions, making them a good place to begin. Lately, there has been a development where certain textbooks created for teachers now provide detailed guidance on how to access and download laboratory materials and additional software that are tailored to their courses. Although the World Wide Web has the necessary capabilities for sharing resources, a higher level of centralized support would be necessary to operate such a system. When an institution solely publishes its content on the Internet, it can provide issues for others in terms of obtaining and validating the quality of the relevant materials. A proficient repository requires the participation of a reviewer to simplify the organization of resources into a more manageable collection and ensure a higher level of quality. It is probable that either an individual, an organization, or an agency will need to assume the responsibility of becoming a reviewer.

Undergraduate Education

The individualization of programs is essential in order to fulfill the diverse educational objectives and components that are associated with different schools. While there are some people who place a higher priority on the practical requirements of the business world, there are others who are more concerned with the more general goals of obtaining a degree in the liberal arts and sciences. As a result of these differences, there is no individual model that can be utilized for all of the different institutions. Some institutions that provide undergraduate degrees in computer science continue to face a wide range of challenges, despite the fact that these challenges vary from institution to institution. A list of the individual issues can be found in the paragraphs that follow. Teaching that is both effective and efficient is not given the same priority as research and faculty fundraising at many universities. It is of the utmost importance that the entire academic community, which includes the teaching staff at both community colleges and universities, reevaluate the relationship that exists between education and research. It is vital that students, educators, and computer professionals establish unique connections between research, education, and industry in order to maximize the benefits of education in a variety of intellectual and practical domains. This is because education can be beneficial in a variety of applications. It is crucial to modify the criteria for tenure and promotion in order to make it possible for faculty members to consider teaching as a pursuit that is equally important and essential to research and consulting.

Therefore, educational institutions need to devise mechanisms to recognize and praise the efforts of faculty members who are developing novel teaching methods and curricula. This is especially important in situations where these initiatives have a substantial impact on other academic institutions. As a result of the rapid rate of development in the field of computer science, professors are required to maintain an active research activity in the subject during their whole careers. Depending on the specific institution and the researcher, this may involve conducting research within established subjects, developing unique instructional materials or curricula based on recent achievements, or investigating unconventional areas such as instructional methods and curriculum. All of these activities are possible. Educational institutions that place an emphasis on instruction ought to lower the number of students they enroll and direct a greater portion of their financial resources on academic endeavors. The provision of chances for continued faculty development through the use of summer seminars, conferences, short courses, and tutorials ought to be feasible. In addition, educational institutions that place a strong emphasis on teaching should provide opportunities for faculty members to participate in apprenticeships in academic contexts that are focused on research and business. Despite the fact that the business sector is the primary employer of graduates in computer science and engineering, many educational institutions fail to take into account the requirements of the business sector. When developing courses for undergraduate students, it is essential to take into account both pedagogical principles and the requirements of applicable industries. The fundamental principles of the area should be the primary focus of the beginner courses, while the implementations of such concepts should be the primary focus of intermediate and advanced courses.

It is essential to introduce the extensive range of themes that are addressed in the subject at an early stage in the educational process. On the other hand, this should not be done at the expense of a solid foundation in programming, which should also involve problem-solving, software engineering, and design. Due to the fact that practical design is not only essential but also powerful, it is of the utmost importance that every student be given the opportunity to participate in it. According to Cupper (1996), it is recommended that all undergraduate applicants be required to take part in either a capstone research experience or a senior-level design course. However, it is also recommended that this requirement be made mandatory. The primary objective of this course is to provide students with the opportunity to gain practical experience in addressing a contemporary research challenge or in locating a possible solution to a significant problem that exists in the real world.The curriculum should begin with an introductory course that lays the groundwork for combining theoretical and practical features throughout the whole instructional program. Semantics in programming languages, verification hurdles in software engineering, concerns in database theory, and the inability to solve algorithms are some examples of problems that can arise.

Furthermore, it is essential for academic staff to remain vigilant for opportunities to reimagine and communicate the conventional topics of the field, including theoretical notions, in a manner that is both original and fascinating. To the greatest extent possible, the experiments that are carried out in the laboratory and the examples that are provided in the textbook ought to address a wide range of student concerns and current events.

When it comes to many educational programs, the management of large classrooms is a significant challenge. In the field of education pertaining to computer science and engineering, lectures alone are not sufficient to meet the educational requirements because they do not include laboratory work or opportunities for students to participate in small groups. It is imperative that there is a greater recognition of the significance of professional computer education and training as an essential component of educational institutions. A significant obligation to provide opportunities for professional development falls on the shoulders of community institutions. Nevertheless, it is recommended that a number of their undergraduate programs offer practical classes to students who are enrolled in evening and part-time classes. This would be in accordance with the different goals of the programs as well as the requirements of the communities that are located in the surrounding areas. It is essential for faculty members to actively seek opportunities to provide lab materials, instructional techniques, and technological tools that are appealing to all students in order to properly cater to the varied learning styles and backgrounds of the students who are enrolled in the Computer Science and Engineering program. Both education and shifts in research specializations ought to be approached with the same opportunistic frame of mind. Experts in the fields of computer science and engineering have been cautious to adopt creative teaching methods, and some of the strategies that we currently use are not helpful in appropriately immersing students who are just starting out in the topic.

It is of the utmost need to develop new curricula that emphasize the interconnection of many different courses, including computer science. Taking the field of computer science as an example, it is feasible to develop interdisciplinary major programs that incorporate the fields of economics, psychology, and the natural sciences within the framework of a computer science concentration. According to Van Loan (1996), the field of computational science is currently going through the beginning stages of its development. When it comes to service courses that are not part of a degree program, significant assistance and ongoing improvement are required. Numerous educational establishments provide inclusive beginning courses that are designed to meet the needs of a diverse spectrum of students, including those who are not majoring in the subject matter as well as those who are. As the discipline of computer science and engineering (CS&E) becomes increasingly complex and prominent, the degree of interest among individuals who are not majoring in the subject has increased. [Kelemen 1996] asserts that it is essential to take into account these different interests when developing new instructional materials and effective ways of instruction.

Graduate Education

There is a wide variety of priorities that are prioritized by educational institutions that offer master's and doctoral degrees, which are comparable to undergraduate programs. In order to conduct effective research, research-oriented Ph.D.-granting universities require their faculty members to stay current on the latest developments in their respective industries. This is the case even though the course offerings may be subject to change. Concurrently, the environment of educational institutions that provide graduate programs is undergoing a transition. The federal government has significantly reduced the amount of money it allocates to research, and there are indications that some additional reductions are likely to take place in the near future. In order for educational institutions to acquire adequate funding for research in the future, it may be necessary for them to form more robust partnerships with other industries. There is a possibility that improvements in technology will result in modifications to the research and instructional objectives of particular graduate programs in the field of computer science and engineering. As a consequence of this, graduate programs in computer science and engineering face a number of specific challenges, which will be discussed in the paragraphs that follow. Instead of allocating persons who have received a Ph.D. degree to academic institutions, the industrial sector will gradually hire individuals who have obtained such a degree of higher education. In the most current Taulbee Survey [Taulbee 1995], it was found that more than sixty percent of newly earned Ph.D. holders are working in the corporate and government sectors, rather than in academic institutions. At the same time, there was a reduction in the number of open positions in CS&E departments between the years 1994 and 1995, and this number has remained unchanged ever since.

As a result of the continued reduction in university funding, it is reasonable to assume that the proportion of individuals who hold Ph.D. degrees and are employed in the academic sector will fall. During the process of developing deeper links with firms, academic departments may decide to make modifications to the courses they offer and the curriculum they follow. Ph.D. applicants, on the other hand, are often given priority by employers for their project management skills rather than their specific expertise in a particular field of study. Because of this, we consider graduate education to be an advantage that departments will continue to enjoy, regardless of whether or not new funding sources are introduced or whether or not stronger corporate connections are established. There are around thirty percent of recent Ph.D. graduates who are employed in departments that do not have graduate programs. On the other hand, only a small percentage of these personnel possess the specific skills and capabilities that are required to properly manage the delicate balance of academic priorities that these departments place a high priority on. Universities are currently placing a larger emphasis on educational activities as a means of producing money. This is due to the limited financial resources that are available for graduate programs, as well as the influence of pressures from both the outside and the students themselves. Taking into consideration the fact that tuition costs for undergraduate students constitute a sizeable amount of most universities' budgets, this is of utmost significance. At this time, the vast majority of Ph.D. programs do not offer adequate financial assistance and infrastructure for the purpose of preparing students for teaching positions and transitioning into other careers. It is imperative that departments make it a priority to teach these skills to a significant percentage of their faculty members as well as their graduate students.

There will be a gradual increase in the market demand for individuals who possess a Master of Science degree in computer science. In spite of the fact that there has not been a recent increase in the number of Ph.D. programs, there is a growing trend of departments offering Master's degrees, particularly in fields where the need is met by the local industry. There is a possibility that this will lead to the rise in popularity of master's degrees that are exclusively comprised of coursework and do not require a thesis or project. Given the diminished prominence of research in the curricula of certain departments, it may be absolutely necessary to reevaluate the equilibrium between research and activities that are implemented in the real world. As a result, it is of the utmost importance that professional education be widely accepted as an integral component of higher education. It is important for graduate programs to provide classes that may be attended in the evenings and on a part-time basis so that they can meet the needs of the inhabitants of the surrounding areas. The professionals and instructors working in the sector who are in need of enhancing or updating their technical competence in creative methods would be the target audience for these courses.

K-12 Education

The following concerns are particularly pertinent to computer training in elementary and secondary schools, despite the fact that some of these principles may also be applicable to the curriculum of universities. At the present time, there is no secondary school curriculum that is recognized on a global scale and is consistent. When it comes to mathematics and scientific sciences, the courses that are taken in high school are often standardized. We are of the firm belief that it is of the utmost importance to give priority to the adoption of a computer science curriculum that emphasizes the fundamental ideas in computing and explains how these concepts can be applied in practical situations at the same time. It is recommended that the ideas that are defined in the ACM Model High School Curriculum [ACM 1993], which offers a comprehensive review of the topic, be adhered to. The current Advanced Placement (AP) program, which was established by the College Entrance Examination Board in 1996, has a more limited focus and does not adequately handle this particular aspect of the curriculum. There are not enough procedures in place to provide teachers with training on the most recent technological advancements or to ensure that they are kept up to date on developments in the industries in which they work. There is a severe shortage of qualified teachers in secondary schools, which is the principal obstacle that prevents computer science education from being provided. In order to effectively address the existing problem, significant efforts are required. In order for new educators to be successful in the field, it is essential for them to develop innovative programs that offer them essential training.

A further objective of these programs should be to encourage graduates of computer science majors to think about pursuing a career in teaching. In addition, it is of the utmost importance to provide educators with substantial training that provides them with the information they need to recognize that computer science and education involves a tremendous amount more than just programming. The training that they receive ought to equip them with the ability to successfully communicate this comprehension to both the general public and their students. There are major distinctions between individuals who have access to technology and those who do not have access to it when it comes to things like technology. There exists a substantial disparity between the educational opportunities that are made available to children in the various school systems. When kids in school systems that are more wealthy are compared to students in districts that are less wealthy, it is clear that the former have significantly more opportunities to acquire technological skills. In addition, even among educators working within the same educational institution, there is a substantial gap between administrators and teachers who are adept in the use of technology and those who are not of the same level of expertise. A great amount of time would need to be committed to the education of teachers and personnel at all levels in order to successfully integrate computer technology into the bulk of the subject areas that are covered in instructional programs. Teachers at the pre-college and college levels are required to develop new ways of communicating with students and providing support. The American Computer Society (ACM) should bring back the undergraduate research award and the high school programming competitions. New initiatives include the creation of a computer science journal for high school students, which is comparable to a math magazine, as well as the provision of new summer courses for both pre-service and in-service secondary school teachers.

Coordination Within the Education Community

This user has provided a list of many committees and organizations that are associated with the education of computer science. The IEEE-CS Educational Activities Board, the Consortium for Computing in Small Colleges, the AP Curriculum Committee, the CRA (graduate education, faculty issues), the CSAB/CSAC (accreditation), EDUCOM (computing and technology in higher education), the ACM SIGCSE (computer science education), the ACM SIGGRAPH Education Committee, the ACM SIGCUE (computer uses in education), the ACM SIGACT Education Committee, and the ACM Pre-College Committee are some of the organizations that fall under this category. The level of interaction between these communities is severely limited, despite the fact that they share a wide variety of interests in common. At this time, the leader of the ACM Education Board is the only person who can make decisions about the selection of new members. The result of this is that only a small portion of the numerous academic initiatives are expressly represented at the top levels of the Association for Computing Machinery (ACM). We are adamant in our belief that the ACM Education Board ought to revise its charter in order to ensure the participation of representatives from all groups and to encourage increased cooperation between different groups. By putting this modification into effect, the board will be able to become more open, inclusive, and prominent, and as a result, it will be able to assume a leading role in the process of improving computer science education as a whole. In order to allow a more productive conversation about issues pertaining to the curriculum, the Education Board had to make an effort to develop innovative communication channels. It is recommended that the SIGCSE program committee include individuals from the graduate school, professional education, and secondary school communities, including the Advanced Placement Program (AP), in addition to the undergraduate community, which is already well-represented. As an illustration of the possible benefits that could be gained from improved communication channels between various groups of individuals, consider the recent decision to switch from Pascal to C11 as the programming language that is utilized in the Advanced Placement course [Abelson et al. 1995].

Creating a Center for Cs &E Education

Assisting each working group in determining "strategic" work directions that would have the most significant medium-term impact on the development of the field was the primary objective of the workshop on Strategic Directions in Computing Research (ACM 1996), which was held in 1996. Consequently, this was accomplished in accordance with the title of the workshop. Our working group has come to the conclusion that in order to guarantee the successful implementation of this criterion in computer science education, it is absolutely necessary to encourage greater resource sharing through the utilization of contemporary computing technologies. The promotion of the reusability of instructional software and upgrades to the curriculum is, in a sense, a big problem for computer science and engineering education. This task is comparable to the ongoing difficulties regarding software reuse in software engineering. The establishment of a decentralized online repository that would aggregate and encourage the sharing of knowledge that is pertinent to the field of computer science education is the goal that we have set for ourselves. The name "CS&E Education Center" is something that we are contemplating for this particular establishment. As outlined below, the CS&E Education Center offers the following services to its clients. the core repository for all of the resources that pertain to the development of courses and the curriculum. A centralized, web-based library of resources to support course and curriculum development would be the primary component of the Computer Science and Engineering Education Center. Examples of such resources would include the publication of lab and course materials, a search and directory function, and a certification procedure that would be supported by an organization such as the Association for Computing Machinery (ACM). In addition to this, the repository would provide a digital forum that would exist continuously for the discussion of topics such as educational programs, interdisciplinary links, and other topics that are relevant. This is a compilation of completed programming assignments and laboratory assessments.

The most challenging component of teaching a course that focuses heavily on programming is coming up with interesting assignments and laboratory activities for students to complete. In the same way that software development requires a significant amount of troubleshooting, writing a high-quality assignment does as well. It is possible to dramatically improve the overall quality of the assignment by incorporating the comments that students provide regarding their experience with earlier revisions. When one distributes a high-quality assignment to other organizations, they are able to optimize the return on their investment. It is possible for teachers at these kinds of educational establishments to search the library for additional assignments that are relevant to their classes and then execute those assignments that have been shown to be the most effective within their particular circumstances. an examination query database that is centralized and a scoring service for the examination. It would be beneficial for a large number of teachers to have access to a single repository of possible test questions, from which they may choose and modify questions for their students. The utilization of a question style that enables online administration and scoring could, under certain conditions, make the process of testing in contexts that are conducive to distance learning more straightforward. Furthermore, the resource center might provide a score mechanism of some form to its potential customers. A collection of instructional resources and tools for giving presentations with multimedia. Students see a considerable gain in their level of knowledge as a result of the utilization of instructional resources that have been thoughtfully crafted and excellent classroom presentations.

As a result of the large amount of time that is required to develop such tools locally, it would be tremendously advantageous for educators in other areas to be able to produce valuable teaching software in order to compensate for the amount of time that is required to construct these tools. Within the context of this field, the CS&E Education Center would perform two different functions. In the beginning, along the same lines as the curriculum repository, the center would take on an editorial duty. This would involve making certain that the resources that were accessible through the repository were of a high enough quality to be employed effectively. Furthermore, the establishment would offer a restricted amount of software support to its customers. Academic institutions and the faculty members who are responsible for developing instructional materials are unable to satisfactorily meet the requirements of a significant number of external clients because of the restricted resources that they possess. A unified resource dedicated to the possible outcomes for students. A primary resource for students who are looking for opportunities to participate in summer research, postgraduate work, fellowships, and graduate school in the field of computing may be the Computer Science and Engineering Education Center. Additionally, despite the fact that this information is already available in a variety of venues, it is frequently not easily understood or arranged. One could refer to it as a "virtual university" that allows for distance learning. It is possible for the center to provide the required technology infrastructure in order to facilitate the delivery of laboratory courses, lectures, and debates through remote interaction. Student-initiated courses (similar to those offered at MIT) with a remote instructor, certified study-from-home programs, virtual faculty or student exchange programs, virtual guest lectures, cooperative coursework among remote student groups, virtual visits to libraries, research facilities, archives, remote internships with industry sponsors, in-service and pre-service courses for secondary school and college instructors at remotely located institutions, and so on are all examples of programs that would be eligible for this program.

Those individuals who are actively employed and have a need for continued education, such as those who are currently enrolled in classes on object-oriented programming, Java, or interactive Web software design, would be considered a target market for these types of courses. Additionally, it would give coverage for the minority of students who are geographically distributed but yet require a course in uncommon academic subjects, such as compiler code optimization and development. This would be a significant benefit. Due to the low demand in the region, it is now difficult for a particular institution to offer such a degree to its students. Those interested in a more in-depth examination of these concepts might visit Rada [1996]. An educational resource that encompasses instructional models from diverse academic disciplines. In the field of computational science, the development of computer-generated models for scientific applications is a crucial area of focus. The curriculum of the Computer Science and Engineering Education Center would encompass a diverse range of subjects and would integrate the concepts, methodologies, and visual depictions utilized in the creation and construction of these models. Computer science courses and collaborative projects between students and professors will be offered to support the development of effective computational models aligned with the research and curriculum goals of the respective fields of study. Examples of virtual activities include creating a model of a financial market, conducting an archeological excavation in a virtual environment, or setting up a virtual laboratory for chemistry. Conducting a substantial amount of exploratory research is essential to effectively tackle this challenge.

A crucial consideration is whether a CS&E Education Center can offer adequate and meaningful human connection to fulfill its educational responsibilities. For example, it is crucial for professionals to swiftly answer students' inquiries and examine the assignments they have received. The government will face substantial pressure from both the public and the industries that will gain advantages from these policies, demanding their approval. Several recent endeavors inside the Electronic Health

Record Directorate of the National Science Foundation are prompting inquiries that are similar to the one currently being explored. An imperative initiative is the construction of a Computer Science and Engineering Education Center, which is characterized by its innovative, intricate, and very important nature. This notion has the ability to ensure the ongoing vitality and significant advancement of computer education in the twenty-first century.

References

ABELSON, H., BRUCE, K., VAN DAM, A., HARVEY, B.,TUCKER, A., AND WEGNER, P., WITH RESPONSE BY HORWITZ, S., APPEL, K., CUPRAK, T., KAY, D., NEVISON, C., SCHRAM, L., STEHLIK, M., AND ASTRACHAN, O. 1995. The first-course conundrum. Commun. ACM, (June).

ABELSON, H., SUSSMAN, G. J., AND SUSSMAN, J. 1985. Structure and Interpretation of Computer Programs. MIT Press, Cambridge, MA. ACM 1968. Curriculum Committee on Computer Science. Curriculum 68: Recommendations for the undergraduate program in computer science. Commun. ACM (March).

ACM 1978. Curriculum Committee on Computer Science. Curriculum 78: Recommendations for the undergraduate program in computer science. Commun. ACM (March).

ACM 1993. Task Force of the Pre-College Committee of the Education Board of the ACM. ACM model high school computer science curriculum. Commun. ACM (May).

ACM 1996a. Workshop on Strategic Directions in Computing Research (MIT, June). http:// www.medg.lcs.mit.edu/sdcr.

ACM 1996b. Special Interest Group on Computer Science Education (SIGCSE). Computer Science Education Links. http://www.ucs.usl. edu/;ram1335/edlinks.html.

ACM/IEEE 1991. ACM/IEEE-CS Joint Curriculum Task Force. Computing Curricula 1991.

ACM Press, New York. Abridged versions reprinted in Commun. ACM (June) and IEEE Computer (Nov.).

ASTRACHAN, O. L. 1996. A Computer Science Tapestry: Exploring Computer Science and Programming with C11. McGraw-Hill, New York (to appear). Preliminary information is available from http://www.cs.duke.edu/;ola/ book.html.

COLLEGE ENTRANCE EXAMINATION BOARD. 1996. Advanced Placement Computer Science Curriculum, Educational Testing Service. CSAB (COMPUTER SCIENCES ACCREDITATION BOARD). 1987. Criteria for accrediting programs in computer science in the United States. Tech. Rep., Jan.

CSTB (COMPUTER SCIENCE AND TELECOMMUNICATIONS BOARD). 1992. Computing the Future: A Broader Agenda for Computer Science and Engineering. National Academy Press, Washington, DC. CUPPER, R. D. 1996. Capstone courses in computer science at Allegheny College. Presented at the 13th LACS Summer Meeting, Dartmouth College, July.

DECKER, R. AND HIRSCHFIELD, S. 1995. The Object Concept: An introduction to Computer Programming Using C11. PWS Publishing Company, Boston.

DENNING, P. J. 1981. Eating our seed corn. Commun. ACM (June). DENNING, P. J. 1984. Educational ruminations. Commun. ACM (Oct.).
DENNING, P. J., COMER, D. E., GRIES, D., MULDER, M., TUCKER, A. B., TURNER, A. J., AND YOUNG, P. R. 1989. Computing as a discipline. Commun. ACM (Jan.).
DIJKSTRA, E. W. WITH RESPONSES BY PARNAS, D. L., SCHERLIS, W. L., VAN EMDEN, M. H., COHEN, J., HAMMING, R. W., KARP, R. W., AND WINOGRAD, T. 1989. On the cruelty of really teaching computer science. Commun. ACM (Dec.).
EDUCATIONAL ACTIVITIES BOARD. 1983. The 1983 model program in computer science and engineering. Tech. Rep. 932, Computer Society of the IEEE, Dec.
FORD, G. 1990. 1990 SEI report on undergraduate software engineering education. Software Engineering Institute Tech. Rep. CMU/SEI90-TR-3 (March).
FREUND, S. AND ROBERTS, E. 1996. Thetis: An ANSI C programming environment designed for introductory use. In Proceedings of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education (Philadelphia, Feb.).
GIBBS, N. E. AND TUCKER, A. B. 1986. Model curriculum for a liberal arts degree in computer science. Commun. ACM (March).
HUFF, C. AND MARTIN, C. D. 1995. Computing consequences: A framework for teaching ethical computing. Commun. ACM (Dec.).
KELEMEN, C. F. 1996. SDCR position statement on the first course for nonmajors. http://www. bowdoin.edu/;allen/sdcr/kelemen.html.
KOFFMAN, E. P., MILLER, P. L., AND WARDLE, C. E. 1984. Recommended curriculum for CS1: 1984 a report of the ACM curriculum task force for CS1. Commun. ACM (Oct.).
KOFFMAN, E. P., STEMPLE, D., AND WARDLE, C. E. 1985. Recommended curriculum for CS2: 1984 a report of the ACM curriculum task force for CS2. Commun. ACM (Aug.).
MULDER, M. C. AND DALPHIN, J. 1984. Computer science program requirements and accreditation—an interim report of the ACM/IEEE computer society joint task force. Commun.ACM (April).
NATIONAL SCIENCE FOUNDATION. 1996. NSF computer science courseware repository. http: //www.education.siggraph.org/nsfcscr/nsfcscr. home.html.
PATT, Y. N. 1996. SDCR position statement on the first course. http://www.bowdoin.edu/ ;allen/sdcr/patt.html.
PROULX, V. K., RASALA, R., AND FELL, H. 1996. Foundations of computer science: What are they and how do we teach them? In Proceedings of the SIGCSE/SIGCUE Joint Conference on Integrating Technology in Computer Science Education (Barcelona, Spain, June).
RADA, R. 1996. SDCR position statement on the Virtual Computing College. http://www. bowdoin.edu/;allen/sdcr/rada.html.

RASALA, R., PROULX, V. K., AND FELL, H. J. 1994. From animation to analysis in introductory computer science. In Proceedings of the ACM Computer Science Conference (Phoenix, AZ, March).
ROBERTS, E. 1993. Using C in CS1: The Stanford experience. In Proceedings of the TwentyFourth SIGCSE Technical Symposium on Computer Science Education (Indianapolis, IN, Feb.).
ROBERTS, E. 1995. The Art and Science of C. Addison-Wesley, Reading, MA.
ROBERTS, E. 1996. Tools for creating portable demonstration programs. In Proceedings of the SIGCSE/SIGCUE Joint Conference on Integrating Technology in Computer Science Education (Barcelona, Spain, June).
RUDICH, S. 1996. CS1: How to think like a computer scientist. http://www.cs.cmu.edu/;rudich.
SHAW, M. 1990. Prospects for an engineering discipline of software. IEEE Softw. (Nov.).
STEIN, L. A. 1996. Rethinking CS101: Innovations in introductory computer programming. http://www.ai.mit.edu/projects/cs101.
TAULBEE 1995. 1995 Taulbee survey. Accessible via the CRA home page: http://www.cra.org.
TUCKER, A. B., BERNAT, A., BRADLEY, W. J., CUPPER, R. D., AND SCRAGG, G. W. 1995a. Fundamentals of Computing I: C11 Edition. McGraw-Hill, New York.
TUCKER, A. B., BRADLEY, W. J., CUPPER, R. D., EPSTEIN, R. G., AND KELEMEN, C. F. 1995b. Fundamentals of Computing II: C11 Edition. McGraw-Hill, New York.
TUCKER, A. B. AND WEGNER, P. 1996. Computer science and engineering: The discipline and its impact. In CRC Handbook of Computer Science and Engineering. CRC Press, Boca Raton, (Dec.) (to appear). http://www. crcpress.com/prods/2909.htm. VAN LOAN, C. 1996. Introduction to Computational Science. Addison-Wesley, Reading,
MA.
WALKER, H. AND SCHNEIDER, G. M. 1996. Revised model curriculum for a liberal arts degree in computer science. Commun. ACM (Dec.) (to appear).