Technology Education for Kids: Cultivating Technology Professionals of Tomorrow and Today

Barbara D. Gannod
Department of Electronics and Computer Engineering Technology
Arizona State University East
7001 E. Williams Field Road, Building 50
Mesa, AZ 85212
Email: bgannod@asu.edu

Abstract

Technology is becoming more and more important in our modern "information age." While engineering, computing, and other technology intensive professionals are in higher demand, universities are finding it harder and harder to attract and retain qualified students in science and engineering programs.[1] A nationally discussed societal problem termed the "digital divide" has a great effect on which students will choose technology-related majors.[2] Other related problems include attracting minorities and women to engineering and science and finding students whose K-12 education provides the background necessary to be successful in science, engineering, and technology-related fields.[1,3,4,5] These challenges, along with the desire to share the excitement of computer science and technology, have lead to the development of the Technology Education for Kids (TEK) program at Arizona State University by undergraduate student members of the Association of Computing Machinery (ACM) and associated faculty.

The TEK program is similar to many other successful programs that bring college students and faculty into K-12 classrooms to teach engineering concepts and mentor K-12 teachers and students; however, the TEK curriculum specifically targets computer science and technology education (rather than general engineering). A preliminary yearlong curriculum was developed by a core group of student members of ASU's student chapter of the ACM under the direction of the author and under the consultation of a local elementary school teacher. The pilot offerings of this computer science-related curriculum occurred during the 2001-2002 academic school year in two different local elementary schools. These initial offerings were very positive both for the elementary school students as well as the college students involved in the program.

In this paper, we describe the TEK program along with its unique goals and contributions to engineering education, and we discuss what was learned during the development and pilot offerings of the TEK curriculum. In particular, we discuss the successes as well as the challenges that were faced. Finally, we discuss the future directions of the TEK program.

1 Introduction

Developing and maintaining strong engineering and science related programs in the university is rife with challenges. A nationally discussed societal problem termed the "digital divide" has a great effect on which students will choose technology-related majors.[2] A related problem that has been widely addressed is the attraction and retention of women and ethnic minorities in science-related majors.[3,4] Another broader problem is finding students that have rigorous pre-collegiate backgrounds that properly prepare them to be successful in engineering and science disciplines.[1] Finally, as technology is becoming more widely available to our society, we are seeing detriments in addition to the benefits that technology brings. Many students are trained to use calculators and computers in their school math courses. Some students become overly reliant on technology and trust that any answer appearing on a calculator or computer screen must be correct; they have not developed the ability to reason about problems, but to trust that technology can reason better than they can. For example, the Mathematics Standards of Learning for Virginia public schools recognizes this as a problem and states, "the use of technology shall not be regarded as a substitute for a student's understanding of quantitative concepts and relationships or for proficiency in basic computations."[6] These challenges, along with the desire to share the excitement of computer science and technology, have lead to the development of the TEK (Technology Education for Kids) program at Arizona State University by undergraduate student members of the Association of Computing Machinery (ACM) and associated faculty.

The primary objective of the TEK program is to make technology accessible to kids. In particular we stress the "how" of technology rather than the "what" of technology. Many schools are teaching students "what" computers can do – word processing, presentations, spreadsheets, database use, Internet searching, etc. (for a typical example, see North Carolina's approved K-12 Computer/Technology Skills Standards).[7] While these are very important skills to obtain in today's digital world, engineers would not consider these subjects to be engineering or computing technology. The TEK curriculum addresses "how" a computer works rather that "what" a computer can do. Thus, we discuss the "science" of computer science and technology. For example, we provide an introduction to computer architecture, discuss how data is stored, and give an overview of algorithm development. In essence, we give an overview of what students would study in college if they were to major in Computer Science, Computer Engineering, or Computer Technology.

While the overarching goal of the TEK program is to make technology accessible to kids, there are several other goals under that umbrella. First, the long-term goal is to develop a curriculum that is presentable to *any* student regardless of whether they have a computer at home, use one at school, or rarely use a computer at all. This exposure to technology is hoped to help in the process of "closing the digital divide." Furthermore, a goal of the TEK program is to take the curriculum to several elementary schools including those with large minority populations that may have students that do not have computers at home or role models that would influence them to pursue computing.

Addressing the other common challenges in engineering and the sciences is part of the vision of the TEK program. Namely, TEK exposes kids to rigorous thinking and problem solving at a young age, with hopes that those kids will acquire and develop critical thinking and problem solving skills that they can use and develop for many years before entering college. The concepts introduced help students with problem solving (breaking a problem down into known input and unknown output) and abstraction (using variables in an algorithm), and thus give an excellent foundation for success in higher-level mathematics. The curriculum also gives students experience with logical reasoning that is important for mathematical proofs as well as general reasoning. Finally, the TEK curriculum is meant to demystify technology. Students are presented with the basic computation model for computer systems. They understand at a basic level how computers work and the concept of "Garbage in, Garbage out." The objective is that students will understand that technology only does exactly what a person tells it to do, and that they must still be able to reason about solutions and check answers against their reasoning.

During the 2001-2002 academic year, a preliminary yearlong curriculum was developed by a core group of student members of ASU's student chapter of the ACM under the direction of the author and under the consultation of a local elementary school teacher. The pilot offerings of this computer science-related curriculum occurred during the 2001-2002 academic school year in two different local elementary schools. These initial offerings were very positive both for the elementary school students as well as the college students involved in the program.

In this paper we describe the TEK program, its goals, and its unique contributions. The remainder of this paper is organized as follows. Section 2 provides a summary of many successful initiatives that aim to incorporate science, engineering, and technology into K-12 education, and it discusses the unique aspects of TEK in this context. Section 3 discusses the TEK curriculum and pilot offerings. In particular, we discuss our successes as well as the challenges that we faced. Section 4 concludes the paper, outlining future directions of the TEK program.

2 Related work

Universities, Industry, State Governments, and the Federal Government are hard at work trying to incorporate engineering education into K-12 schools. An Internet search on "K-12 Engineering Education" results in about 340,000 hits! Creighton provides an excellent survey of various approaches to K-12 engineering education.[1] She describes three approaches that have been used: "Top-down approach, Franchise Model, and the Hub-and-Spoke Method." We believe we could also divide the programs into two basic categories: (1) those that send industry personnel or university faculty and students into schools to help with engineering instruction and mentoring [8-17] or (2) those that create well-design curriculum that can be given or sold to schools for K-12 teachers to use.[18, 19] Some programs include aspects of both categories. The two approaches have different advantages, but both approaches have been successful. The major advantages of sending industry mentors or university faculty and students into the K-12 classrooms are that the technical experts are teaching the material. The experts are more likely to be able to answer tough questions and adapt the material for individual classroom interests. Additionally, the college students provide excellent role models for the K-12 students. The major advantage of having the K-12 teachers present the material developed by technical experts is that K-12 teachers are trained to present material at an appropriate level – they are the t*eaching* experts rather than the *technical* experts.

The number of university programs that address K-12 science and engineering education is too large to enumerate here; Sanoff [2] describes many successful K-12 engineering education initiatives that are representative of the countrywide effort to incorporate engineering principles into K-12 classrooms. Most of the programs aimed at K-12 engineering education address "Engineering" proper – electrical engineering, mechanical engineering, civil engineering, etc. Some programs contain a unit on computer engineering or robotics or some other computing subject. For example, the Center for Engineering Educational Outreach at Tufts University has developed curriculum for programming LEGO Mindstorm robots.[8] However, very few (if any) of the programs specifically target computer science and computing technology. In a draft report, the ACM K-12 Education Task Force Computer Science Curriculum Committee describes the current status of K-12 computer science education.[20] Most of the current computer science curriculum has been developed for high school students. Some schools offer an AP Course in computer science that emphasizes problem solving, algorithm development and programming.[21] The ACM Task Force proposes a K-12 computer science curriculum that builds on previous efforts.[22,23] In the proposed curriculum, simple computing such as keyboarding, searching the web, algorithmic thinking, and interactive programming is done at the primary level (K-8) to prepare students for more rigorous study of computer science. In grade 9 or 10 topics such as computer organization, more advanced algorithmic problem solving, discrete mathematics, logic, and networks are covered. Additional courses primarily focused on programming and software engineering would follow (grades 10 or 11-12). This curriculum is still in draft form and has not been widely implemented.

There seems to be a stark difference between K-12 *Engineering* Education and K-12 *Computer Science* Education. Industry Engineers, Professional Societies, and Universities are developing *engineering* curriculum and sending *engineering* experts into K-12 classrooms. Professional Societies and Universities are proposing *computer science* curriculum, but do not seem to be sending *computer science* and technology experts into the classroom to teach and mentor K-12 students. The TEK program developed at Arizona State University shares the vision of those currently making efforts to introduce core computer science concepts in the K-12 curriculum, and the TEK program is eager to emulate the many successful programs that have been developed to teach and excite K-12 students about engineering.

We believe the TEK program is innovative and unique in a couple of ways. First, the TEK curriculum focuses solely on computer science and technology; it does not address general engineering. Second, the faculty and students associated with the TEK programs go into K-12 classrooms to teach and mentor students in computer science and technology. Finally, many of the core computer science concepts (e.g., computer architecture, binary representation, digital logic, algorithms and complexity) are introduced at a younger age than is outlined in the K-12 curriculum proposals that have been developed. Specifically, the TEK curriculum is designed to be used in the fourth to seventh grade range, and can be modified to accommodate either end of that spectrum.

3 TEK pilot offering

During the summer and 2001-2002 academic year, student members of ASU's student chapter of the ACM along with the author and a local elementary school teacher developed curriculum that consists of a set of hands-on lessons meant to excite and teach students about computer science and technology. The curriculum is broken into two semester modules: the first semester curriculum is meant to provide students with an overview of computer science, and the second semester curriculum is designed to give students in-depth experience with a specific topic in computer science and technology. The curriculum was designed this way so that K-12 schools could choose either a 1-semester or 2-semester option. In this section, we describe the preliminary first and second semester modules that were created, and we discuss the successes and challenges faced during the preliminary offerings.

3.1 First semester TEK curriculum

The first semester TEK curriculum was designed to provide an introduction to core computer science topics. The curriculum consists of a number of weekly lessons (labs) covering the history of computing, basic computer organization and architecture, data representation, and algorithms.

The first few lessons introduce computer architecture. After completing the labs, students understand that a computer is made up of four major components (input, output, memory, processor) that must communicate via a system bus in order to accomplish work. We emphasize the difference between input/output and input devices/output devices. In addition, students understand the fetch-decode-execute role of the microprocessor and can conceptualize memory size and processor speed. Students look under the cover of a personal computer and relate the hardware that they see to the four major components that they have learned. They also look inside a diskette and discuss how diskettes work.

A single lab on the history of computing familiarizes students with the development of the computer. Students think about why the computer was developed and recognize that computing has changed drastically since its inception.

Algorithms and the notion of complexity are covered in the next several labs. Students learn the definition of an algorithm and write algorithms for every day tasks with which they are familiar. They are able to make assumptions about input and then create an algorithm with control structures to produce the correct output. Through these exercises, students recognize that there is more than one method to solve a problem. This leads to the introduction of the notion of algorithm complexity. Students learn that time is connected to the number of steps performed, and that an algorithm that uses fewer step will take less time.

The next set of labs covers how computers represent and operate on data. Students learn that computers store data using binary numbers that represent standard characters (via ASCII), and they are able to convert between decimal and binary representations. In addition, they learn an algorithm for converting decimal numbers to any base, and they can perform addition in other bases. The conclusion of this set of lessons introduces the basic logic operators: AND, OR, and NOT. Students understand that computers must use logical operations to accomplish anything, and they learn how logical operations are used to perform binary addition.

The intent of the final lesson is to tie together the major concepts covered during the semester in a real-world example. A LEGO Mindstorms programmable robot is demonstrated. Concepts of input and output are reviewed as students are asked to determine the inputs given to the robot and the outputs produced. Algorithms are reviewed as the students determine the algorithms the robot must use to go forward, stop, turn, follow a line, etc.

3.2 Second semester TEK curriculum

The second semester curriculum is meant to build on the first semester curriculum, but narrow the focus to a single topic. The culmination of the second semester is a significant project that allows the students to apply their knowledge in a "real-world" setting. The second semester curriculum of the pilot offering covered the design and implementation of digital circuits. At the end of the semester, students were broken into small groups, and each group was given a design problem. The students designed the circuits, tested them via simulation, and finally implemented their solutions in an electronics laboratory during a field trip to Arizona State University's East Campus.

The second semester curriculum prepared the students to design and implement digital circuits. Students reviewed the binary number system and basic logic design. Basic Boolean algebra and circuit minimization techniques (K-maps) were covered. The students practiced the techniques by walking through the design process for a binary adder and a seven-segment display. The solution for the seven-segment display was implemented via a hardware simulation tool to verify its correctness. Finally, the students were given a problem for which they were responsible to design and implement their own solution. The design problems were similar to problems that college students would be given in their first logic design course.

Figure 1 TEK student implementing project in electronics laboratory

Figure 1 shows a TEK student working on her final project. Her problem description was as follows:

Design a digital circuit for controlling a process heater and process cooler. The heater (or cooler) can be either "on" or "off", however, both the heater and cooler should NEVER be on at the same time. A thermostat detects when the process temperature is above or below the desired temperature. You may assume that the thermostat has value "0" if the temperature is above desired and has value "1" if the temperature is below desired. Both the heater and the cooler are turned off when a pressure gauge detects an overpressure condition. However, there is a manual override switch that allows the heater or cooler to operate even when there is an overpressure condition.

3.3 TEK pilot offerings and lessons learned

When the TEK curriculum described above was developed, a fifth grade level was the target. A local elementary school teacher was consulted to ensure that the material was attainable at that level, and that the methods used to present the material were appropriate. The fifth grade level was chosen for a couple of reasons. First, many experts agree that we need to reach kids while they are still young – long before college or even high school. Jackie Sullivan, the co-director of the Integrated Teaching and Learning Program at Colorado College of Engineering was quoted as saying engineering should be presented by at least the third grade.[1] There was also a very practical reason for choosing fifth grade: we had a teacher to work with and specifically had her class of gifted fourth and fifth graders in mind for our initial offering of the material.

The pilot offerings of the TEK curriculum occurred during the 2001-2002 academic year. The first semester curriculum was presented to a small class of gifted fourth and fifth grade students during the Fall semester of 2001. The first semester curriculum was refined and presented to a class of sixth graders during the Spring semester of 2002 while the second semester curriculum was given to the initial group of fourth and fifth grade students.

The initial results were very positive. Pre and Posttests and a final review for the first semester curriculum via a "jeopardy"-type game showed that the students had learned the concepts that were covered during the semester. Successful implementations of the digital design projects by all groups during a field trip to an electronics laboratory verified that students were able to apply their study of digital electronics in the second semester to a "real-world" problem. The majority of the elementary school students seemed to be excited about the material that they were learning, and it was exciting to watch the college students as they developed a deeper understanding of computer science as they worked on the lessons and techniques to present the material in a low-level, but meaningful way.

A significant lesson learned is that kids from ages 10-12 can learn many of the computer science core concepts that are typically left until much later to learn. While the ACM K-12 Computer Science Task Force advocates incorporating computer science into all K-12 grades, they leave topics such as principles of computer organization, binary numbers, digital logic, and algorithm complexity/efficiency for the high school years .[20] While fifth graders probably cannot understand these subjects at the same level as high school students, they are certainly able to grasp these concepts enough to have a basic understanding of the "big picture" of computing. With the background that the TEK curriculum provided, students would be very well prepared for a deeper study of computer science in high school.

While the pilot offerings of the TEK program have been very successful, they were not without challenge. One problem that we have faced is attracting and retaining undergraduate students to develop and present material. Students involved with the TEK program are strictly volunteers – they are not paid a stipend. In order to have consistent help with the program, fellowships and the rewards and responsibilities that come along with a paid position would be very beneficial. Another challenge that was faced was classroom discipline. The college students presenting material in the elementary school classrooms were computer science majors – not education majors. The kids often viewed and treated them as “student teachers,” but the computer science college students do not have the same kind of training that a student teacher would have before entering a classroom. Consequently, the college students found themselves frustrated at times. This is why some programs have decided to develop material to give or sell to K-12 teachers and to leave the teaching to the K-12 teaching experts.[18]

4 Conclusion

The Technology Education for Kids (TEK) program developed by ACM student volunteers and supporting faculty and its initial offerings to fourth through sixth grade students has been rewarding and instructive. The TEK curriculum was developed as a two-semester sequence of lessons, where the second semester is optional. The major objective of the curriculum is to demystify computing technology and to generate interest and enjoyment in the science and engineering that is behind one of the most influential advances of the past fifty years. The first semester provides an overview of computer science. It includes topics such as history of computing, computer organization and architecture, the binary number system and digital logic, and algorithms and algorithm complexity. The second semester is meant to be an in-depth study of a specific area of computer science or technology. Digital logic was the second semester topic of study in the pilot offering. The culmination of the second semester was a field trip to an electronics lab where students used TTL parts to implement their design of a solution to a real-world problem.

The TEK program is still in its infancy. The first semester curriculum continues to be refined. It is being presented to a new K-12 school in 2002-2003. Additionally, new second semester modules need to be developed. Possible second semester topics include data structures, programming, and networking. Furthermore, more rigorous assessment mechanisms need to be created to evaluate the benefits of the program. In order to accomplish many of the goals of the TEK program, it needs funding to support the students and faculty that develop, present, and assess material.

Bibliography

1. L. Creighton, “The ABCs of Engineering,” *ASEE Prism*, vol. 12, no. 3, pp. 23 – 27, 2002.

2. A. P. Sanoff, “Closing the Digital Divide,” *ASEE Prism*, vol. 10, no. 8, pp. 17-21, 2001.

3. M. Mannix, “Getting it Right,” *ASEE Prism*, vol. 10, no. 7, pp. 14-20, 2001.

4. E. Seymour, “Revisiting the ‘problem iceberg’: Science, Mathematics, and Engineering Students Still Chilled Out,” *Journal of College Science Teaching*, vol. 24, no.5, pp. 392-400, 1995.

5. A. Sanoff, “Building Tomorrow’s Workforce,” *ASEE Prism*, vol. 10, no. 6, pp. 16-22, 2001.

6. Mathematics Standards of Learning for Virginia Pulic Schools, http://www.pen.k12.va.us/VDOE/Superintendent/Sols/mathsol2001.doc

7. North Carolina's State approved K-12 Computer/Technology Skills Standard Course of Study, http://www.dpi.state.nc.us/curriculum/computer.skills/

8. http://www.ceeo.tufts.edu/default.asp

9. S. Poole, J. DeGrazia, and J. Sullivan, "Assessing K-12 Pre-Engineering Outreach Programs," *Journal of Engineering Education*, vol. 90, no. 1, pp. 43-48, 2001.

10. J. DeGrazia, J. Sullivan, L. Carlson, and D. Carlson, "A K-12/University Partnership: Creating Tomorrow's Engineers," *Journal of Engineering Education*, vol. 90, no. 4, pp. 557 – 564, 2001.

11. http://itll.colorado.edu/ITLL/

12. J. Batty and K. Batty, "Bringing the K-12 Stakeholders onto the Engineering Education Team", *Proceedings of the 2001 ASEE Annual Conference & Exposition*, 2001.

13. http://www.usu.edu/stc/

14. http://depts.washington.edu/k12smt/index.html

15. http://www.k12science.org/ (Steven's Institute of Technology)

16. http://tc.engr.wisc.edu:16080/zwickel/outreach/

17. http://www.usfirst.org

18. http://www.infinity-project.org

19. http://www.enc.org

20. A. Tucker, F. Deek, J. Jones, D. McCowan, C. Stephenson, and A. Verno, draft of "A Model Curriculum for K-12 Computer Science: Report of the ACM K-12 Education Task Force Computer Science Curriculum Committee," Nov., 2002. http://www.acm.org/k12/

21. AP Course Description: http://www.collegeboard.com/ap/students/compsci/index.html

22. Task Force of the Pre-College Committee of the Education Board of the ACM, "ACM Model High School Computer Science Curriculum," 1997. http://www.acm.org/education/hscur/

23. F. Deek and H. Kimmell, "Status of computer science education in secondary schools," *Computer Science Education*, vol. 9, no. 2, 1999.

BARBARA D. GANNOD
Dr. Gannod is an Assistant Professor in the Department of Electronics and Computer Engineering Technology at Arizona State University East. She completed B.S. degrees in Mathematics and Computer Science and a Secondary Education degree at Calvin College in 1992. She received her Ph.D. in Computer Science from Michigan State University in 1997. Her research interests include engineering education and distance learning.