AC 2007-1176: THE EFFECTS OF PRIOR COMPUTER EXPERIENCES IN CONSIDERING ENGINEERING STUDENTS' ABILITY TO SOLVE OPEN-ENDED PROBLEMS

Christian Hipp, University of South Carolina

Veronica Addison, University of South Carolina

The effects of prior computer experiences in considering engineering students' ability to solve open-ended problems

ABSTRACT

This paper relates one part of a National Science Foundation (NSF) funded, exploratory research project in the Course, Curriculum, and Laboratory Improvement Program (CCLI). The research project's objective is to determine the best ways to introduce computing into early undergraduate mechanical engineering curriculum, focusing particularly on numerical methods and analysis. Given the importance of computing in professional engineering practice, this project seeks to improve students' facility with computers while moving away from 'cookbook' approaches which emphasize software-specific skills at the expense of more fundamental mathematical and conceptual knowledge.

One aspect of this research project was to determine what computer experiences (STEM -- Science, Technology, Engineering, Mathematics -- or otherwise) students have when they enter college-level engineering classes. We surveyed sophomore engineering students in "Introduction to Applied Numerical Methods" (EMCH 201). Along with a freshman course in Graphics and Visualization, Numerical Methods constitutes one of the two engineering courses in which beginning students first use computers for engineering applications. The technical platform in EMCH 201 is a symbolic manipulator, currently Mathcad.[i] This survey attempted to gauge both their previous experiences with computers and their assumptions about how to use them, what computers do well, and when, whether, and why they trust the results provided by computers. The surveys focused on three issues: prior computer experience, student assumptions about computers, and extent of their trust in computer-generated solutions.

INTRODUCTION

Arguably, our students have lived their whole lives in the era of everyday computing. Consequently, when they take engineering classes in college, professors are hardly 'introducing' them to computers; rather, they are showing new applications. Thus, engineering educators need to understand better the pre-existing experiences and assumptions students have about computing, and exploit those tendencies for better comprehension of engineering concepts whether computer-aided or not.

To think that students' cookbook expectations come solely from engineering classes is naïve – 'cookbook' approaches also dominate software instructional methods. From the time a student first loads a piece of software onto a computer and first uses a help menu, she becomes inculcated into a cookbook approach to software. Furthermore, students' previous experiences with computers are a particularly crucial issue for a public university where students enter with a wide variety of education exposure to computers.[ii]

In professional practice, the great power of the computer often stems from its flexibility in facilitating a multitude of solution techniques. However, in teaching engineering students to use the computer, that flexibility is often hard to communicate—students are inclined to think of the computer as a needlessly more complicated version of their calculator. Therefore, we need to find better ways to articulate pedagogy with professional engineering practices using computers.

The larger research project, of which the survey discussed here is but one piece, focuses on finding ways to efficiently develop skills that will enable students to solve more complex problems as well as get students to think more creatively about computer-aided approaches and to see problems with 'cookbook' or 'plug-and-chug' approaches for themselves.

SURVEY, DATA, AND RESULTS DISCUSSION

Data were collected through a survey instrument (see below). Fifty-six engineering students from a required sophomore-level numerical methods class (EMCH 201) completed the survey at the beginning of their sophomore year, to document their prior exposure to computers and evaluate how the engineering curriculum both challenges and reinforces students' prior assumptions.

The surveys attempted to measure students' previous experiences with computers and their assumptions about how to use them, what students thought computers did well, and when, whether, and why they trusted the results provided by computers. The first two tables display results about their computer experience for STEM applications and more generally. The final tables of questions sought to determine the assumptions students have about the ways computers are used and the kind of problems they can solve and to establish the extent to which students place unreasonable trust in computer-generated solutions.

TABLE 1. HOURS/WEEK OF COMPUTER USE

1. On average, how many hours each **week** do you currently spend on computers (word processing, gaming, internet, email, using various software, etc.):	**0**	**1-5**	**6-10**	**11-15**
	0%	3.5%	25%	28.5%
	16-20	**21-25**	**26-30**	**31+**
	21%	12.5%	2%	7%

Computer use appears to be arguably a normal distribution, with the mean in the 11-15 hour category. No one surveyed indicated no time on the computer. This latter claim suggests the ease and availability of computer access. Averaging a few hours a day, students do not spend exorbitant amounts of time using a computer.

TABLE 2: COMPUTER USE

2. Prior to college, how often did you utilize a personal computer at the following places:	**ALWAYS**	**OFTEN**	**SELDOM**	**NEVER**
a. At Home?	61%	34%	5%	0%
b. At School?	9%	38%	52%	1%
c. At Library?	5%	23%	48%	23%
d. At internet café?	1%	0%	5%	94%
e. other: _______ (*only one other category added – "Work"*)	7%	0%	0%	93%
3. Since college, how often do you utilize a personal computer at the following places:				
a. At home?	41%	32%	21%	6%
b. Dorm room?	36%	12%	40%	48%

c. At apartment?	48%	16%	4%	32%
d. Computer lab?	18%	42%	36%	4%
e. School library?	0%	18%	62%	20%
f. Public library?	0%	0%	36%	64%
g. Internet café?	1%	1%	0%	98%
h. Other? ______ (*only one other category added – "Work"*)	3%	3%	0%	94%
4. I use the computer for:				
a. Internet	75%	21%	2%	2%
b. Gaming	20%	32%	41%	7%
c. Word Processing	30%	62%	8%	0%
d. Instant Messaging	44%	30%	17%	9%
e. Music or video downloads, edits, etc.	45%	36%	17%	2%
f. Graphing, designing, or drawing	6%	30%	52%	12%
g. Homework or academic projects	43%	46%	11%	0%
h. Programming	0%	3%	52%	45%
i. 'Day Timer' (planner, address book, etc.)	1%	11%	22%	66%
j. Data processing	6%	17%	37%	21%
k. Reading books, magazines, journals, periodicals, etc.	12%	30%	37%	21%
l. Calculating or Problem-solving	12%	21%	61%	6%

Prior to and since college, most used computers at home – assuming 'home' is where they live most of the time (possibly including 'dorm' and 'apartment') as distinguished from parent's home. The largest gain appears to be in the computer lab. Generally, computers were predominantly used for internet, word processing, homework/academic projects, and music/video downloads. Arguably, the top three uses could overlap in academic areas. Students indicated using the computer for problem-solving and data processing; this indication is interesting and surprising since most engineering classes that may require this kind of work are at the junior and senior level.

TABLE 3. ASSUMPTIONS & TRUST

STATEMENT:	STRONGLY AGREE	AGREE	DISAGREE	STRONGLY DISAGREE
1. Computers could solve humanity's moral problems.	0%	20%	63%	17%
2. I have found computers to be more reliable than people are.	5%	39%	50%	6%
3. Someday computers will do all the work.	9%	39%	43%	9%
4. I trust computers.	7%	71%	21%	1%
5. A computer is more like a machine than a mind.	39%	59%	0%	2%
6. Based on their past performance, software processes are predictable.	18%	66%	14%	2%

7. Computers can only solve scientific or mathematical problems.	7%	50%	39%	4%
8. Computers should take over only mundane repetitive tasks.	14%	45%	38%	3%
9. Computers are always right.	2%	10%	61%	27%
10. More functions in society should be given over to computers.	7%	34%	55%	4%
11. A computer is more like a typewriter than a calculator.	0%	4%	64%	32%
12. I rarely check the solution of a problem from a computer program I'm familiar with.	20%	45%	32%	3%
13. Computers are smarter than people.	0%	16%	64%	20%
14. Most people do not need to learn about computers in school.	0%	7%	29%	64%
15. I trust a human assistant more than I trust software processes.	11%	55%	34%	0%
16. Computers should not put people out of work.	39%	36%	23%	2%
17. Search engines' success is more determined by search parameters than the data searched.	16%	77%	7%	0%
18. Computers should be designed to support the tasks of human users.	27%	70%	3%	0%
19. Computers think like humans do.	0%	5%	45%	50%
20. Computers rarely give faulty information.	0%	37%	54%	9%
21. Computers are only a kind of tool.	28%	61%	9%	2%
22. Highly automated systems need constant monitoring.	18%	73%	9%	0%
23. I would trust the solution given by artificial intelligence.	0%	57%	39%	4%
24. A computer-generated solution is always correct if the data is entered correctly	6%	50%	40%	4%
STATEMENT:	**STRONGLY AGREE**	**AGREE**	**DISAGREE**	**STRONGLY DISAGREE**
25. Software programmers are more responsible for the success or failure of an application than software users.	13%	45%	40%	2%
26. The correctness of a computer-generated solution is more determined by the data entered than the program	7%	68%	25%	0%

utilized.				
27. Software designers do not know what users need.	5%	9%	73%	13%
28. A computer is more like a friend than a foe.	12%	86%	2%	0%
29. In my experience, computers are dependable across a range of operations.	18%	80%	2%	0%
30. Computers always do what they are told.	11%	34%	50%	5%
31. Sometimes computers act totally bizarre.	20%	59%	20%	1%
32. Trust in software processes is derived from earned trust in the software programmer.	2%	84%	14%	0%
33. A computer is used more for fun than for work.	3%	38%	59%	0%
34. Artificial intelligence has the potential to endow computer systems with human-like capabilities, such as judgment, planning, and problem-solving.	6%	41%	46%	7%
35. Computers are complex.	30%	52%	16%	2%

Generally, students surveyed have particular assumptions about the kinds of problems computers can solve and tasks to which computers ought to be applied. While there is a general split among students assuming that computers can only solve mathematical and scientific problems, most do not think computers can solve moral problems. Thus, students seem to assume there are kinds of problems computers cannot or should not solve. This evidence may suggest an underlying view of knowledge that may need correction. Moral and scientific problems are not so neatly divisible. Claims about what counts as a legitimate problem, solution, evidence, and the like are made in the context of normative assumptions.

Most students think computers should not put people out of work and overwhelming espouse that computers should be used primarily in support roles. Yet, there was a general split about whether computers will do all the work in the future, that computers should not be given more things to do, and that computers should only take over mundane repetitive tasks. Students may be indicating what they prefer or desire as contrasted with what they think will likely occur. Furthermore, this evidence suggests possible contradictions in the ways students think– *e.g.*, 75% thought computers should not put people of work but approximately 40% thought more functions should be delegated to them. This suggests that students are not rationally reflexive about their assumptions and dispositions regarding computers.

Regarding learning about how to use computers, most students think that computers are complex and that people do need to learn about computers in school. Students may be indicating an openness to more education content about what computers can do and what applications they may apply computers to in their curriculum. This evidence may also suggest an openness for engineering educators to have more instruction about computers' capacities through specific applications as well as more open-ended kinds of problems. Generally, students appear familiar with computers, assume that computers can do things that they do not yet do with them, assume a healthy caution about the level of computer use, and comfortably view computers as machines.

Regarding trust dispositions, only a slight majority find people more reliable than computers. Most think that past performance suggests reliability, which may relate to why most students do not check the answer that a computer gives them. Furthermore, students overwhelming find computers dependable in many cases. However, a majority of students do not trust computers, still trust a human assistant more, assert that computers are not always right, and think that computers give faulty information. As noted in paragraphs above, this evidence appears to suggest contradictions in student dispositions and assumptions – computers are dependable but not trustworthy; humans are less reliable but more trustworthy. This evidence may give warrant to the moral and epistemological dimensions of trust (i.e., trust involves moral norms and evidential norms) and that trust is not reducible to reliability. Additionally, students do not appear to be strongly disposed to the source of the error that warrants their skepticism – only a slight majority thinks error is mostly due to data input. Yet, a slight majority does not think that computers always do what they are told, while a majority of students think that computers can sometimes act totally bizarre.

Generally, the splits that do exist on matters of trust do not entail an abnormal resistance to technology but a healthy awareness of the danger to trust any 'thing' too much. However, students do not appear to be sure why or where the demarcation line should be drawn. This evidence may suggest the problem of students 'black-boxing' the computer's operation (what, how, and why of computer's operation is not relevant to know in order to use and apply it), which may even account for the mixture of reliability and mistrust. Incidentally, this finding is additionally interesting because this 'blackboxing'/trust phenomenon is common among those who are not technologically adept (*i.e.*, 'I don't know how it works so I don't know how or whether to trust it'). Furthermore, this issue is apparent among engineering students who are technophilic as a general rule, perhaps indicating a need to know more about how things work in order to wield judgments about trust in computers. Are 'computer experts' closer to the 'general public' than we might have thought? Or, is this condition possibly a general characteristic of sophomores that improves by the senior year? In either case, the problem is likely a correctable, curricular one.

Students clearly have a high level of familiarity with the idea and presence of computers. They readily accept computers as part of life. Students do not appear to be blind followers or resist the rate of change. These general conclusions point to a pedagogical strategy our project may implement in some degree – to design more cases where the computer gives the 'wrong' answer (a student entered data expecting one answer and given another, problem given that is intentionally poorly designed, giving a problem in which no actual solution exists, etc.).

CONCLUSION

The next step in the larger research project will be to evaluate the effects of timing and order in the way Numerical Methods is taught. The timing and order of presentation of concepts and skills is a critical component in engineering classes. Often distinctions are made between a theory and its application to a particular problem. Computers can be used to demonstrate both theory and application, but in either case increase the complexity of establishing an effective order of presentation. The survey will also be used to track possible changes in perspective of these students by giving the survey at the end of the Numerical Methods course. The survey will be used to assess engineering students at different levels in the curriculum to compare, for example, freshman and senior engineering student perspectives about the computer. Additionally, students in non-engineering disciplines (*e.g.*, humanities) may be surveyed to further offer comparisons for analysis.

ACKNOWLEDGEMENT & DISCLAIMER

This material is based upon work supported by the National Science Foundation under Grant Number 0536660 -- "CCLI: Effects of Timing, Content and Instruction on Engineering Students' Abilities to Solve Open-Ended Problems using Computers." All opinions expressed within are the authors' and do not necessarily reflect those of the National Science Foundation.

ENDNOTES

[i] Our research questions are not software specific and would be equally served by MatLab or Mathematica. Also, the concepts introduced include consideration of programming languages independent of these packages (*e.g.*, Visual Basic, Fortran, or C).

[ii] This issue could be exemplified at the University of South Carolina by the contrast between students coming from technologically well-equipped urban school districts and under-resourced rural school districts.