Brief and Yet Bountiful: The History of Computing, Why Do Students Need It?

Cuihua Zhang
Computer and Information Science
Northwest Vista College
3535 North Ellison Drive
San Antonio, Texas 78251
Voice: (210) 348-2182
E-mail: czhang@accd.edu
Web: http://www.accd.edu/nvc/areas/cis/bio/zhang.htm

John E. Howland
Department of Computer Science
Trinity University
One Trinity Place
San Antonio, Texas 78212-7200
Voice: (210) 999-7364
Fax: (210) 999-7477
E-mail: jhowland@Trinity.Edu
Web: http://www.cs.trinity.edu/~jhowland/

April 17, 2005

Abstract

This paper argues that a separate history of computing course at the upper division of a liberal arts computer science program is necessary and helpful. The study of the history of computing will provide correct and insightful perspectives for students as to how technology and science have evolved into what they are today. It is also likely that students will be enthused and consequently make greater contributions to the discipline. Students will have the opportunity to learn not to make the same mistakes that their predecessors made, and avoid the detours that our predecessors took. This paper also provides a brief research summary on how the subject is being dealt with today. [1]

Subject Areas: Computer History, Computer Science Education, Computer Science Curriculum.
Keywords: Computer History.

[1]This paper is published in the Journal of Computing Sciences in Colleges, Volume 20, Number 4, April 2005, Pages 308-314.

1 Introduction

History is what has made us who we are today. Human beings have always tried to learn from history. History has taught, bettered, encouraged, and enlightened us. The history of computing is not long, and yet, dense with events and turning points that have readied it for study. Indeed, there are many studies that have been done or are being done. However, this paper proposes that a separate history of computing course be offered as an upper division course. Such a course will definitely intrigue as well as enlighten students. Who knows how many future science and technology virtuosos amongst our students will be awakened and inspired by the lives of those in the past? From Charles Babbage to John Von Neumann, to Bill Gates, from the Analytical Machine to UNIVAC, to today's PCs, the span of the history of computing is replete with enthralling stories and awe-striking inventions. If we do not tell students about them, we are missing a wonderful opportunity to educate them. This paper explores current history of computing courses, delineates the advantages of such a course offerings, states at what level we should offer history courses and explores possible topics.

2 Current Research and Resources

Depending on how we look at computer science, the history of computing is both long and short. It is long in the sense that the earliest calculator, the abacus, was invented in China in the 14th century [9]; it is only sixty years according to Mollenhoff [14] and Burks [2], both of whom called Atanasoff the father of the computer, the person who invented the first electronic digital computer during 1937 to 1942. The history of computing was not mentioned in the official Computer Curriculum until 1991 when the Association for Computing Machinery (ACM) and IEEE Computer Society Joint Task Force explicitly included it in some areas [10]. Computer Curricula 2001 [4] designated history of computing as SP1: Social and Professional Issues, with one lecture hour. With both the imperative-first and object-first model illustrated in Curricula 2001, history of computing was covered in CS111, an introductory course to computer science or programming.

A random sampling of colleges and universities (From the alphabetically ordered list of universities and colleges [5], the 10th institution in each lettered list is sampled) showed no schools have a course titled History of Computing. The website maintained by Martin Campbell-Kelly [3], last updated October, 2003, shows a list of 20 universities and colleges around the globe that offer history of computing as a full-scale course, and not all of them list this course as a CS course. Some of them offer it in the history department. There might be some universities or colleges that are offering history courses but was not on this list. Tim Bergin, who currently serves as Editor-in-Chief of the IEEE Annals of the History of Computing journal, teaches at American University CSIS 550, History of Computing, a one semester course. Nevertheless, the number is likely to be very small. The institutions that do offer history of computing as a CS course either deal with it as social, cultural, and professional issues or delve into the technical aspect of the history, for example, discussing the Turing machine and how it works, etc. Resources for the study of history of computing are not lacking. The International Federation of Information Processing (IFIP) Working Group 9.7 (WG9.7), established in 1992, aims at providing the "central vehicle" for the study of the history of computing [8]. Since 1980, there are more books related to the subject [10]. Books and Journals can be easily found. There are quite a few museums and libraries, both physical and virtual that are devoted to this topic. Some examples include the Computer History Museum [17], the Virtual Museum of Computing [1], etc. The IEEE Annals for the History of Computing is a publication that publishes four issues each year starting 1979 (two issues in 1979). The full text can be viewed when logged onto the ACM/IEEE digital library. In the United Kingdom, there is the National Archive for the History of Computing, which is the UK's most important resource for history of computing [15]. There are scholars who keep websites devoted to the subject also, for example, Thomas Haigh has one that maintains a Computer History File [7].

However, as was pointed out by Lee [13], among all the resources, it does seem that there is not a

comprehensive book that can be used as a textbook. Lee, a lead scholar of the history of computing, thinks that it is time that a comprehensive text book on the history of computing be written, which can be used throughout the curriculum. Teaching professionals could use this book to teach the history of their respective area.

3 Why Students Need a Computer History Course

Why is history of computing often not available in educational institutions then? The 1965 prediction by Gordon Moore that microprocessor chip density would double every 18 months has proven to be true continuously. With the rapid change of hardware components, networking has increased tremendously, and the World Wide Web has exploded. "Over the last decade, computer science has expanded to such an extent that it is no longer possible simply to add new topics without taking others away."[4]. With new technologies coming, one followed immediately by another, it is naturally assumed that the present and future of computer science demand more attention; and therefore, the history, which happened already and could not possibly be changed at all, is ignored.

History is a record of our successes as well as our mistakes. Looking back at the mistakes and studying them helps us avoid making the same or similar kind of mistakes. We may have learned some lessons at a certain point; we may have forgotten some of them; or we may have ignored some. Those that are ignored or forgotten "are in danger of being repeated,"[10] and that would be a heavier price to pay.

When we teach students good programming practice, we usually tell them "spaghetti code" is bad; we tell them object-oriented design is good, etc. We teach them with such high definiteness, it almost sounds religious [18]. However, if we show them some "goto" statements used in the past, if we tell them how Allan Kay invented Small Talk, the first computer language based entirely on the notions of object and messages, then our lecturing would not sound so high-tone and religious; and students would develop a real appreciation for the axiom we teach them. By going through the process of how that definiteness was achieved, we will be stimulating the students thinking [18] and students will believe what we teach them, not because we tell them so but because they understand the reasons behind and the historical perspectives. By studying the history, students also learn the motivations [10], the circumstances, and the environment under which an invention came about [11]. Seldom do things come uncalled for; decisions are made for a reason. Inventions and creative ideas are nurtured by the social environment and students should know this. By teaching and discussing the motivating conditions, we are providing them the ability to recognize certain conditions and needs and therefore, we are really nurturing their creativity.

Some of our students will be researchers, scientists, and engineers, but some of them will become educators. Knowing the history will enable them to intertwine their lectures with little anecdotes that will enliven their classrooms and intrigue their students. Offering a history of computing course will also fill in the need for writing across the curriculum [12]. Computer science students are usually dubbed with the stereotype that they only know how to write code, but do not know how to write an essay. If this is not too far away from the truth, then as educators, we need to provide our students with the very necessary skills of reading and writing and give them the opportunity to practice those skills with subject matter relevant to their major. By having them read many materials on computing history, by assigning them papers to write about the social climate around certain inventions and changes, and about certain pioneers, we are training the basic skills that they need to succeed in the real world.

4 Lessons Learned

As an example of a lesson one can learn from the study of computing history, Andy Hertzfeld, one of the designers of the original Apple Macintosh computer, described one of his major design mistakes [6] in the

design of the memory management software for the original Macintosh operating system. During the time that the Macintosh was being developed, in the early 1980s, the cost of memory system components was relatively high. This limited the memory size of the original Macintosh computer to 128K. The Motorola 68000 processor had 32bit address registers; but, in spite of Moore's law, Andy decided to use the most significant byte of memory addresses to hold memory manager information such as memory segment lock bits. The 68000 processor did not use the high byte during actual addressing operations and Andy predicted that memory sizes would not grow quickly enough for this design choice to be a problem.

However, within three years, Macintosh II computers with a 68020 processor could easily have more than 16MB memories and the memory manager had to be redesigned to be what Apple referred to as "32 bit clean". It took more than a year for Apple software engineers to identify and eradicate all uses of the high byte of addresses in the operating system and applications to fix this problem.

If Andy had studied computing history, he might have learned that IBM made a similar mistake in the design of the System 360 computer during the early 1960s and Digital Equipment Corporation in the design of the PDP 11 [16] series of mini-computers. The high byte of System 360 addresses was used for specific purposes and IBM did not anticipate how quickly memory system sizes would grow. This design choice made it difficult to design compatible large memory versions of the System 360. The DEC PDP 11 systems used 16 bit memory addresses. Later versions, PDP 11/45 and PDP 11/70, used 18 bit and 22 bit addresses. These later designs did not add enough extra memory capacity and also posed compatibility problems when designing software to run on any machine in the PDP 11 family of computers. Different design choices may have been made had the designers, in each case, had knowledge of earlier design mistakes and a proper perspective for the life of their designs in relationship to how quickly competitive forces within the computing industry continue to validate Moore's law.

Operating systems designers face this same question today when they consider how large address pointers should be when porting an existing operating system to 64 bit machines. It is tempting to use pointers which are less than 64 bits because it may be some time before 64 bit machines will actually have 10^{19} byte memories. For example, Intel manufactures about 80,000,000 Pentium IV chips each year and if each machine had, on the average, a one GB memory, that is $8x10^{16}$ bytes of memory for all those machines. This is three orders of magnitude smaller than a single memory of size 10^{19} bytes.

5 History at What Level?

Computing history could be intertwined with the material being taught, from introductory courses to advanced courses. However, there are not many textbooks in the various areas of computer science that provide historical perspective and the ever-increasing core content of our courses does not allow extra time to cover the history of each subject being taught. Curricula 2001 [4] designated History of Computing as a one lecture topic on social and professional issues, and most schools cover it in an introductory course in the first year. However, history of computing, besides being a social and professional issue, is also a science issue, a technology issue, an issue about the past as well as the future. It is also about the reasoning of why certain mistakes are made, and the awakening to how we can do things better in the future.

When we teach history of computing within an introductory course, we can only tell our students a chronology of events and some of the historical figures. We are only showing them a "sketch book", which, given that today is an information age, they can easily find and "flip through" on their own. To teach it as an introduction can excite new students interest, and may recruit more students into the program; nevertheless, it will fail its more important responsibility: teaching students about past mistakes so that they can avoid repeating them, and help them explore the interrelationships among the different subjects in the area to gain a more comprehensive understanding of the discipline. If history does not teach about these choices and intertwined relationships, then history would have lost its meaning.

The authors feel that a separate course on the history of computing is the proper way to address the

history of our discipline, particularly the interrelationship of core topics. For example, computer architecture and operating systems have influenced each other from a historical point of view. Progress in memory technology has influenced the design of computer graphics display devices, etc. Technology developments have often been determined by the ability of a company to market the technology resulting in a change in the history of computing determined not by the technology itself, but rather by marketing skills and business environment. Separate history courses are the norm in other disciplines such as mathematics, art, English, and chemistry, and they are most often offered as upper division courses. The reason is that to learn from the history of a discipline, students have to have certain knowledge about the discipline first. It is felt that history of computing courses should be upper division courses because students who have completed the lower division core courses are in a better position to understand the historical environment of key design decisions from a technical point of view. For instance, students should be well prepared in programming, data structures, computer organization, and theory before a serious study of computing history. Ideally, an in-depth computing history course, taught at the senior level, could provide the opportunity of integrating diverse topics and illuminate the common underpinnings of mathematics, available technology, and other factors such as marketing and business environment which have made our field an exciting place to work. An upper-division computing history course will help students in their choice of capstone project.

6 History Course Objectives

Each core discipline within computer science has its own history. For example, there is the history of programming languages, hardware, networking, databases, etc. A computer history course must provide the opportunity for a serious historical treatment of the interrelationships of these topics. Memory technology had a profound influence on computer organization and databases, but less of an impact on programming languages or networking.

The objectives of the study of computing history should include:

- To provide the development of computing within the context of social, scientific, technological, and business environments
- To connect computing history to other disciplines such as economics, sociology, science, mathematics, and technology
- To provide in-depth treatment of the history of computing within the core areas of hardware, software, theory, and applications

7 Conclusions

History is more than a sequence of events, more than the birthdates of the pioneers, more than who invented the first computer. It is about why and how the sequence of events happened, and the consequences of those events; it is about the historical mistakes that were made and the valuable lessons they teach; it is about providing the perspective for understanding the decisions of the past that led to the present we now experience. These perspectives will aid our students as they make the choices that determine the future of our field. History of computing deserves more than a brief survey; it merits a three credit hour, upper division course that studies it. It will not only tell about the geniuses that invented today, but rather, provide the insight that it may not always be possible to determine a clearly recognizable path from the past to the present. When we search for heroic figures whose works give the complete story on a topic, we must remember that such people may not have existed. We must determine the societal influence on each important work to ensure that our presentation of computing history does not distort our picture of the present and future.

A course of history of computing within the computer science curricula, particularly within a liberal arts setting will provide a rich opportunity for reading and writing, which is often missing within our discipline; it will enlighten our students, encourage our students, excite our students and enable our students to learn the valuable lessons, avoid making the same mistakes, understand the discipline more comprehensively, and face more choices as to what they want to do in their future careers.

References

[1] Bowen, J., The Virtual Museum of Computing (VmoC), http://vmoc.museophile.com/, 2004.

[2] Burks, A. and Burks, A., The First Electronic Computer: The Atanasoff Story, Ann Arbor, MI, The University of Michigan Press, 1998.

[3] Champell-Kelly, M., Courses in the History of Computing, http://www.dcs.warwick.ac.uk/~mck/HoC_Courses.html, 2004.

[4] Computing Curricula 2001, http://www.computer.org/education/cc2001/report/, 2000.

[5] Conlon, M., American Universities, http://www.clas.ufl.edu/CLAS/american-universities.html, 2003.

[6] Hertzfeld, A., Folklore: The Original Macintosh, http://www.folklore.org/index.py

[7] Haigh, Tl, Computer History File, http://www.tomandmaria.com/tom/Resources/ResourceFile.htm, 2004.

[8] History of Computing, http://www.comphist.org, 2004.

[9] 1000 BC to 500 BC: the Invention of the Abacus, http://www.maxmon.com/1000bc.htm, 2004.

[10] Lee, J. A. N., "History in the computer science curriculum", ACM SIGCSE Bulletin, 28, (2), 15-20, 1996.

[11] Lee, J. A. N., "History in the computer science curriculum: part II", ACM SIGCSE Bulletin, 30, (2), 11-13, 1998.

[12] Lee, J. A. N., "History in computer science education: across the curriculum initiatives", ACM SIGCSE Bulletin, 32, (4), 9-10, 2000.

[13] Lee, J. A. N., "Where did history go?", ACM SIGCSE Bulletin, 34, (4), 11-12, 2002.

[14] Mollenhoff, C. R., Atanasoff, *Forgotten Father of the Computer*, Ames, IA: Iowa State University Press, 1988.

[15] National Archive for the History of Computing, http://www.chstm.man.ac.uk/nahc/, 2004

[16] The PDP-11 FAQ, http://www.village.org/pdp11/faq.html, 2004.

[17] Where Computing History Lives, http://www.computerhistory.org/, 2004.

[18] Williams, M. R., "Do we teach computer science as religion or as history?", ACM SIGCSE Bulletin, 32, (4), 4-5, 2000.