Bringing Relevance to Computing Courses through History

John Impagliazzo, Mohammed Samaka

Bringing Relevance to Computing Courses through History

John Impagliazzo
Emeritus, Department of Computer Science, Hofstra University, Hempstead, New York USA
John.Impagliazzo@Hofstra.edu

Mohammed Samaka
Department of Computer Science and Engineering, Qatar University, Doha, Qatar
Samaka.m@qu.edu.qa

Abstract: This paper shows ways in which computing history can make the delivery of teaching computing courses relevant. The authors' approach involves using computing history as a recurring theme throughout courses by adapting relevant historical stories or material to enhance course delivery and to capture student interest. The use of computing history often makes a positive and constructive improvement in courses by making them more interesting, stimulating, and thereby, informing students with non-technical elements in their computing specialties. This approach to computing studies should prove to be a helpful addition to student studies and provide them with a stronger understanding of the computing field in their careers.

Keywords: Computing history, history in computing courses, computing history and relevance

1 Introduction

Literature shows that computing history could be an effective pedagogical tool to teach computing courses. History contributes to students' lifelong learning experiences and it encourages them to appreciate the field. History also enables students to gain a better sense of the nature of inquiry, the processes of innovation, and the human dimension [1]. History enables students to explore beyond machines and expand their view on ways in which computing affects society [2]. Historical diversions from basic course material could be simple stories used to enhance student learning. Teachers could use computing history as a vehicle for extra credit or topic enrichment, or for students who enjoy the softer side of the computing field.

Embedding history in computing courses often depends on the initiative of the instructor. In general, computing teachers never studied such history in undergraduate or graduate courses unless they enrolled in a course through their own interest or initiative.

The use of history in courses depends mainly on the initiative and efforts of the individual teacher. Teachers often need to bootstrap themselves and make some effort to incorporate historical topics in their courses. However, the rewards derived from such engagement are satisfying and they outweigh the cost of the efforts involved.

Our experience shows that further exploration on the use of history in computing and related courses is a worthwhile and engaging endeavour.

2 Relevance and Computing History

We believe that history enhances the teaching of computing. Instructors should use history as a vehicle to enrich specialized course studies and to derive other benefits for learning. One reason is the many adages of describing the necessity of understanding and heeding the past because we may be doomed to repeat it in the future. Indeed, the statement by George Santayana that states, "*Those who cannot remember the past are condemned to repeat it*" [3] already exists within the fabric of the computing field. Stories abound on repeating the mistakes of prior happenings; many are well known. Can we explain the cause for the downfall and ultimate demise of Control Data Corporation (CDC)? What caused Digital Equipment Corporation (DEC) to fail? In the mid-2000s, IBM surrendered its personal computer business to Lenovo. Why did IBM do this when "Big Blue" was synonymous with computing machines?

Answers to such questions are quite complex and we do not attempt to answer them here. Nevertheless, they do form a good starting point for student interaction in a classroom setting and stimulate new ideas and concepts. For some students, they may even provide a basis for more exploratory study and research. Computing history also provides fodder for curiosity and intrigue. For example, as far back as 2001 Microsoft started developing its Surface computer, which became a reality as a coffee table in 2004 and formally unveiled in 2007 [4]. Yet, other companies exploited that technology in the manufacturing of tablets and smart phones in the latter part of that decade. Exploring reasons why a company developed a technology while competitors exploit it could be an effective way to engage students in a topic and promote sound (and even heated) class discussion.

Teachers should dispel the notion that the focus of history applies only to past events. We can make history futuristic. Developing new technologies on the successes of others is useful for students to know in a modern age. It is even more useful to avert the mistakes of others, assuming we know those mistakes. Having an inclination or understanding of computing history is one way to avoid such mistakes. Students should view history as an asset rather. Making computing history relevant to a field of study can enable students to treat it as an asset in conjunction with the technical topics they learn.

Professionals (practitioners and teachers) in the computing field have often ignored the history of their field. They often favour cutting-edge approaches and disregard the adage stated earlier, namely, that those who ignore history are doomed to repeat it. While some organizations have contributed to history preservation, the contributions are isolated and incomplete. Specialized textbooks on technical subjects often do not contain sections on the history of the subject, and when they are included, unfortunately the content often contains myths and inaccuracies. However, in recent times, resurgence is developing where greater awareness of computing history is making strides. Websites such as the IT History Society [5] and conferences such as those held by the Society for the History of Technology (SHOT) [6] address the importance of the topic.

As mentioned before, teachers of specialized computing courses are not likely to have the formal preparation in computing history. However, sufficient resources are currently available and are sufficient to include threads or themes of history in the computing courses they teach. They only need a desire to enrich their class sessions to be successful. Our experience leads us to believe that their students will not regret the enrichment.

3 The Importance of Storytelling

Presenting any history as a series of chronological facts is counterproductive. Storytelling (not gossip or folklore or random rambling) can enrich a subject while connecting students with human elements. We have found that historical storytelling can make computing courses more interesting. Moreover, the computing information and knowledge embedded within these stories are often relevant to some aspect of their own lives. Hence, students tend to remember stories and the computer content within them.

As teachers, we need to adopt more innovative and interesting approaches toward student learning if we expect to achieve student success. Berkeley professor and famed computing theorist Christos Papadimitriou stated [7]:

> *This narrative mode [storytelling] of thought is fundamental for at least two reasons: First, narrative richness is an essential precondition for the self (the converse is, of course, trivial: there can be no narration without narrator). We think of ourselves almost exclusively in terms of our mental autobiography. Second, stories are in a certain intrinsic sense interesting, in that they are attractive, high-priority memory fodder. Everything else being equal, we are much more likely to remember a story than a logical argument.*

There is much truth in this statement. Students, especially those not specializing in computing, are more likely to remember a story about computing than a particular fact about it.

Storytelling can be quite effective. However, teachers should not expect that using this technique is always easy. Stories should evolve naturally. They must be relevant and relate to the topic under discussion. Sources for such stories could emerge from current happenings or events such as a newspaper article, or they might build on some historical background such as a book on computing history. Whatever the source, teachers should make an effort to research historical topics so the stories they tell are meaningful to the topic. We have found that storytelling associates students with different perspectives on computing and it connects them with realistic and relevant things such as people, places, and events. We definitely encourage a dynamic involvement by teachers with an active participation by students to make historical storytelling a memorable component to all computing courses.

4 History in Computing Courses

We now provide a brief illustration on ways in which computing history can complement and enrich some of the computing courses found in a computing curriculum.

4.1 History and Computer Architecture

Computer architecture courses are one area where the history of the subject has received attention. Many textbooks in this area often contain sections, appendices, or sidebar vignettes that contain images of the machines that support the architecture under discussion. Students often find such attention devoted to history interesting and revealing. Introducing students to computers without discussing their “invention” leaves a void in their education and a lost opportunity to question the inspiration of a few among many. We often start such a course with the question, “Who invented the computer?” and we tend to admonish those students who say, “Bill Gates” or “Steve Jobs” in response. A homework activity surfaces and names such as Babbage, Turing, Atanasoff, Zuse, and Mauchly emerge. A few students “get hooked” into the intrigue and before long, they develop an interest in computing history.

The question of the invention of the computer is complex and is not within the scope of this narrative. Nevertheless, it does provide a good starting point to engage students in seeking answers to the question. It also diffuses discussions from an emphasis on “firsts” and changes the focus on strategies used to process data. The discussion easily shifts to ways people or teams designed those machines to achieve the computation of data, not only from the early days, but also from modern approaches to these strategies.

It is natural that computers become part of computing history since computing machines are tangible artefacts and have easy recognition by most people, even to non-specialists in computing. Additionally, as teachers, we wonder sometimes whether our students have ever seen the internals of a computer and its electrical and electronic elements such as resistors, capacitors, flash memory chips, busses, or motherboards. More importantly, do these computing specialists know the manner in which these elements work? Notwithstanding, an overview of the history of the subject would naturally expose such elements at least through images if not through visits to computing museums such as the Computer History Museum in California [8], the Science Museum in London [9], or the Deutsches Museum in Munich [10].

4.2 History and Computer Networks

Another area of computing where history has received some attention is computer networks. One related area is communication networks. The internet as we know it has propelled much of this attention. Making this topic relevant in a modern communications course adds meaning and interesting knowledge. For example, do students know what the word “internet” means? Actually, we could trace the root of the internet to 1957 when the Former Soviet Union successfully launched Sputnik [11]. Students might ask: what does Sputnik have to do with the internet?

Students should reflect on world history and recall the "cold war" threat that existed for decades since 1945, the end of the World War II. In response to Sputnik's launch, the United States formed the Advanced Research Projects Agency [12] in 1958 within its Department of Defense (DoD). The purpose of ARPA was to establish a military advantage in science and technology, since a satellite launch from a seemingly adverse country posed a national security threat.

By 1969, ARPA created the first internet with only four nodes as appears in Figure 1. They called the network ARPAnet. The nodes were located at the:

1. University of California at Los Angeles (UCLA)
2. Stanford Research Institute (SRI)
3. University of California at Santa Barbara (UCSB)
4. University of Utah

Each of the four nodes had a specific function such as special processing or graphics. Together, they operated as a whole and were able to process data, even though they were many hundreds of miles away from each other.

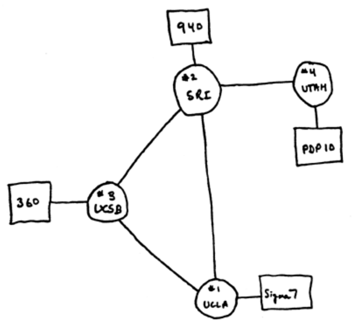

Figure 1: Classical sketch of the four-node ARPAnet from 1969
Source: http://www.computerhistory.org/internet_history/full_size_images/1969_4-node_map.gif

It is our experience that students find stories such as this one very fascinating. They actually provide fodder for further exploration on topics such as the following.

- How many nodes are on the internet today? How does one count such nodes? Who does the counting?
- How does the internet compare with the world wide web? Are they two entities of the same thing? If not, how do they differ? Can one exist without the other?

These and other related questions lead to class or interpersonal discussions that open up new knowledge based on knowledge students already have. We have found that this approach adds relevancy and enrichment to a topic or subject and makes network or communications classes more inviting.

We could explore other historical episodes that relate computing history to technical topics in a networks or communications plan of study. The outcome would be similar or the same. Our experience shows that using the approach of history or story telling *captures* the imagination of students and makes the teaching of the technical aspects of a course more enjoyable and interesting to students and teachers. Students will definitely explore questions such as those above; teachers could use the results of student findings and exploration as a part of their marks or grades for a technical course (e.g., 5% to 10%) or as extra credit, which students always enjoy receiving for work beyond the expectations of a syllabus.

4.3 History in Other Specialized Courses

Teachers can use computing history in other subject areas where teachers can use stories as a vehicle to generate some excitement and interest. However, the "show and tell" strategy can become limited. With integrated networks and "system on a chip" (SOC) technologies in today's world, it is difficult to demonstrate physical entities compared to demonstrations of valves or vacuum tubes, transistors, floppy disks, and the like that we could do with computers from the past. Yet, even though an entire computer system is in the shape of a small geometric entity or cube, it is possible to establish interesting stories on the evolution of a machine such as the ENIAC that occupied a large room to modern versions of SOCs. We could do the same for specialized areas such as computer graphics or robotics.

The strategy of show and tell becomes limited as we move away from tangible computing entities to intangible ones. What would one show physically about software other than a package or a download of a language product or an operating system? Yet their history remains intangibly rich. We have encouraged students to trace the roots or ancestry of a modern language today such as C#. When students dig into this, they begin to find interesting connections. The same is true with operating systems. Teachers can pose interesting questions such as what is the status of COBOL and FORTRAN languages or why did the IBM OS2 operating system die.

Of course, many charts and timelines exist on tracing the roots of computing languages and operating systems. We have marvelled at the way students enjoy the fascination of seeing ways in which different languages emerged. Additionally, we see that students do appreciate exploring the relevancy of past languages to the languages in use today. With operating systems, student response to learning historical connections is similar. Here again, an optional assignment on a relevant topic would definitely enrich a class related to software.

Occasionally, we have observed that a student begins to develop a keen interest in the study of the origins of programming languages and operating systems. In such cases, we have directed them to communities of special interest such as SIGPLAN of ACM or to some specialized literature of books. Of particular note are the proceedings of two conferences, namely, the "History of Programming Languages" (HOPL) in 1978 [13] and in 1993 [14].

In our experiences, the use of "show and tell" seems to fall short with theory topics. Here, the focus is often on individuals and their circumstances. With topics related to discrete mathematics, algorithms, data structures, or computer theory, we often take the opportunity to explore the life or achievements of the individuals involved. An obvious individual in computer theory is Alan Turing, whose contributions to code breaking, machine learning, and the virtual Turing machine are well known. We have found that students begin to gravitate and to appreciate individuals for their contributions.

The field of computing has a long list of historical contributors such as Leonhard Euler, John von Neumann, and Maurice Wilkes in addition to more contemporary individuals such as Bill Gates, Steve Jobs, Mark Zuckerberg, Larry Page, or Sergey Brin. Whether centuries ago (e.g., Euler), decades ago (e.g., von Neumann), or contemporary pioneers (e.g., Gates), students relate and become interested in the way their subject of study evolved over time and how individuals shaped and reshaped the computing field. They also see the relevancy of the way lives of computing pioneers affect their own paths to their careers. As teachers, we find this student experience fulfilling.

4.4 History and Introductory Courses

Is history useful in introductory computing courses?

We are in the affirmative on this question. In addition to introducing the field of study, these courses often serve as vehicles to support students' interest in a subject area and develop some understanding of the subject's concepts and fundamentals. Introductory courses are often of an overview nature or of a specialized nature such as an introduction to programming. Overview courses tend to include students from diverse specialties; in specialized introductory courses, the majority of students include those with their intended specialty.

Although useful in either setting, we have found that the use history works best in the overview course. The broad scope of the overview course shows students that the field of computing is more than a narrow orientation. In fact, it is more about an exploration of computing where teachers can easily tailor history into their syllabus. The overview course presents many opportunities for students to explore extensions of topics on their own. We have found that history is a perfect way to relate new concepts with people, places, and events. Students appreciate historical excursions and they can relate their finding to relevant topics in their contemporary settings. As before, students also appreciate receiving extra credit for reporting on their findings. Furthermore, if we integrate history in introductory courses, students would expect its appearance in subsequent courses, which would encourage other faculty members to embrace the historical perspective.

4.5 History beyond Technical Courses

History can also be of relevant interest in courses beyond those of a technical nature. One such course is computing ethics or other similar name. Discussion of contemporary issues such as intellectual property rights can form a basis for further explorations such as when did the concept of intellectual property first begin. Computing ethics has emerged as a required topic of study and it is often part of a computing curriculum as an individual course or as an integrated theme.

It would be good to have students explore real case studies associated with computing and identify ethical situations or dilemmas. Unfortunately, repositories of historical case studies are very limited or isolated and they are generally not accessible to the public. Perhaps this is primarily the result of teachers and business entrepreneurs being overly protective of the sensitivity of those projects that were successful or in particular, those that were unsuccessful. Nevertheless, it is important for students to experience real situations such as the lawsuit of Apple v. IBM and ascertain whether any ethical transgressions were at issue in addition to legal ones. Experiences such as these are very relevant to students' understanding of the computing filed as they start their journeys toward their professional careers.

5 Conclusion

In summary, it is our experience that the inclusion of history with specialized computing topics and courses adds a sociological dimension to the subject. This strategy has had the effect of engaging students in these subjects; they also appreciate the relevance of discussions that include people, places, and events. Furthermore, it has the attribute of enriching a topic or course and providing students with an avenue for extra credit as allowed by the instructor.

Although not discussed explicitly in the narrative above, teachers and students should avail themselves of the many online resources available to them such as virtual museums, "walk-through" galleries, and oral histories that provide deeper descriptions and understandings of specialized historical events. A resource such as the *IEEE Annals of the History of Computing* [15] contains an oasis of formal articles, memoirs, anecdotes, and obituaries; it also provides a foundation for many stories about computing distributed across many areas of the field.

Once again, we have found that using history to teach computing courses is a useful endeavour. The rewards derived from the *teaching* experience certainly offset the preparation and the effort. The rewards derived from the *learning* experience should leave a lasting impression long after students complete a course. Such aspects of engagement add relevance to the subjects students learn as they prepare themselves for professional careers.

References

1. Impagliazzo J., Campbell-Kelly, M., Davies, G., Lee, J.A.N., Williams, M. (1999), History in the Computing Curriculum. IFIP TC 3 / TC 9 Joint Task Group, *IEEE Annals of the History of Computing*, Vol 21, No 1, pp. 1-15, January 1999.
2. Impagliazzo J., (2005), "History: A Vehicle for Teaching Introductory Computing Courses". Proceedings, IFIP 2005 World Computer Conference in Education (WCCE), Cape Town, South Africa, 4-7 July 2005.
3. Santayana, George. (1905) George Santayana (1905) *Reason in Common Sense*, volume 1 of *The Life of Reason*
4. Microsoft, Surface Table. http://money.cnn.com/2012/06/19/technology/microsoft-surface-table-pixelsense/index.htm
5. IT History Society. http://www.ithistory.org/

6. Society for the History of Technology (SHOT), Annual Conference 2013, http://www.historyoftechnology.org/annual_meeting.html
7. Papadimitriou, C. (2003), "MythematiCS: In Praise of Storytelling in the Teaching of Computer Science and Math", Invited Editorial, ACM SIGCSE Bulletin, Vol. 35, No. 4, pp 7-9.
8. Computer History Museum. http://www.computerhistory.org/
9. Science Museum, London. http://www.sciencemuseum.org.uk/
10. Deutsches Museum–Munich. http://www.deutsches-museum.de/
11. National Aviation and Space Administration (NASA). "Sputnik: The Fiftieth Anniversary". http://history.nasa.gov/sputnik/
12. DARPA History. http://www.darpa.mil/About/History/ARPA-DARPA__The_Name_Chronicles.aspx
13. Wexelblat, Richard L., 1981, The History of Programming Languages, Academic Press, 758 pp.
14. Bergin, Thomas J. and Richard G. Gibson, 1996, The History of Programming Languages - II, Addison-Wesley Pub. Co., 864 pp.
15. *IEEE Annals of the History of Computing*. http://www.computer.org/annals. Accessed 2013 July 23.