Evolution of Computing

James R. Larus

Abstract Computers and computing emerged within a lifetime and completely changed our world. Although their history is brief, the change they precipitated has been rapid and constant. Today's world would be unimaginably different without these machines. Not necessarily worse, but certainly slower, static, disconnected, and poorer. One has to look back to the steam engine in the nineteenth century or electricity in the early twentieth century to find technologies with similar rapid and far-reaching effects. This chapter briefly describes the evolution of computing and highlights how its growth is closely tied to concerns of digital humanism.

1 Introduction

Electronic digital computers have existed for only 75 years. Computer science—or informatics, if you prefer—is roughly a decade older. Computer science is the expanding discipline of understanding, developing, and applying computers and computation. Its intellectual roots were planted in the 1930s, but it only emerged in the 1940s when commercial computers became available.

Today's world would be unimaginably different without these machines. Not necessarily worse (computers emerged during but played little role in the world's deadliest conflict), but certainly slower, static, disconnected, and poorer. Over three-quarters of a century, computers went from rare, expensive machines used only by wealthy businesses and governments to devices that most people on earth could not live without. The technical details of this revolution are a fascinating story of millions of peoples' efforts, but equally compelling are the connections between technology and society.

Like the emergence of a new animal or virus, the growth of computing has serious and far-reaching consequences on its environment—the focus of this book. In seven decades, computing completely changed the human environment—business,

J. R. Larus (✉)
School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland
e-mail: james.larus@epfl.ch

H. Werthner et al. (eds.), *Introduction to Digital Humanism*,
https://doi.org/10.1007/978-3-031-45304-5_3

finance, social relations, government, and society, to name a few—through its seminal advances such as personal computers, the Internet, the World Wide Web, mobile computing, machine learning, and artificial intelligence. One has to look back to the steam engine in the nineteenth century or electricity in the early twentieth century to find technologies with similar rapid and far-reaching effects.

This chapter offers a brief overview of the evolution of computing and its connection to the concerns of digital humanism. The velocity and broad impact of computing's emergence discussed in this chapter partly explain why the humanism implications discussed in the rest of this book are among society's prominent and concerning challenges.

2 Prehistory

In most people's opinion, computer science started in 1936 when Alan Turing, a student at Cambridge, published his paper "On Computable Numbers, with an Application to the Entscheidungsproblem" (Turing, 1937). This paper settled a fundamental open question in mathematics by showing that a general technique does not exist to decide whether a theorem is true or false.

More significantly for this history, Turing's paper introduced the concept of a universal computer (the Turing Machine) and postulated that it could execute any algorithm (a procedure precisely described by a series of explicit actions). The idea of a computing machine—a device capable of performing a computation—had several predecessors. Turing's innovation was to treat the instructions controlling the computer (its program) as data, thereby creating the infinitely malleable device known as a stored program computer. This innovation made computers into universal computing devices, capable of executing any computation (within the limits of their resources). Even today, no other field of human invention has created a single device capable of doing everything. Before computers, humans were the sole universal "machines" capable of being taught new activities.

In addition, by making computer programs into explicit entities and formally describing their semantics, Turing's paper also created the rich fields of program and algorithm analysis, the techniques for reasoning about computations' characteristics, which underlie much of computer science.

A Turing Machine, however, is a mathematical abstraction, not a practical computer. The first electronic computers were built less than a decade later, during World War II, to solve pressing problems of computing artillery tables and breaking codes. Not surprisingly, Turing was central to the British effort at Bletchley Park to break the German Enigma codes. These early computers were electronic, not mechanical like their immediate predecessors, but they did not follow Turing's path and treat programs as data; rather they were programmed by rewiring their circuits.

However, soon after the war, the Hungarian-American mathematician John von Neuman, building on many people's works, wrote a paper unifying Turing's insight

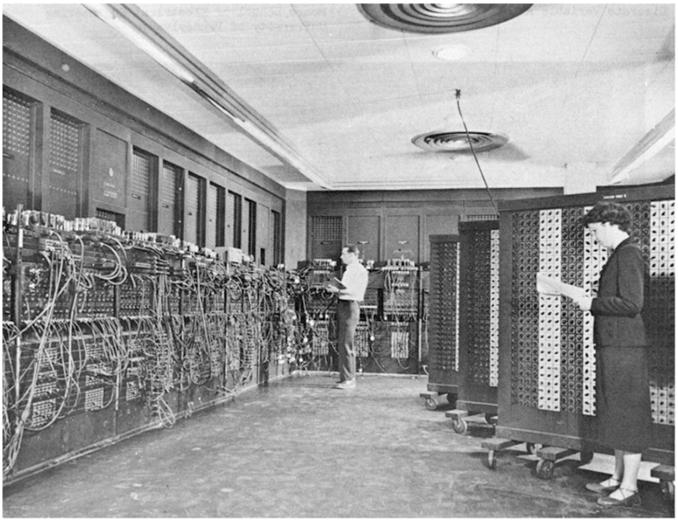


Fig. 1 ENIAC (1947). (Public domain) In Wikipedia. https://en.wikipedia.org/wiki/ENIAC

with practical engineering. It described an architecture for stored-program computers, which laid the computer industry's foundation. This so-called von Neuman architecture still is the blueprint for today's computers. Figure 1 shows a picture of ENIAC, the first general-purpose electronic computer.

3 Computers as Calculators

The first applications of computers were as calculators, both for the government and industry. The early computers were expensive, slow, and limited machines. For example, IBM rented its 701 computers for $15,000/month for an 8-h work day (in 2023 terms, $169,000) (na, 2023a). This computer could perform approximately 16,000 additions per second and hold 82,000 digits in its memory (na, 2003). While the 701's performance was unimaginably slower than today's computers, the 701 was far faster and more reliable than the alternative, a room full of clerks with mechanical calculators.

The challenge of building the first computers and convincing businesses to buy them meant that the computer industry started slowly. Still, as we will see, progress accelerated geometrically. The societal impact of early computers was also initially small, except perhaps to diminish the job market for "calculators," typically women who performed scientific calculations by hand or mechanical adding machines, and clerks with mechanical calculators.

At the same time, there was considerable intellectual excitement about the potential of these "thinking machines." In his third seminal contribution, Alan

Turing posed the question of whether a machine could "think" with his famous Turing Test, which stipulated that a machine could be considered to share this attribute of human intelligence when people could not distinguish whether they were conversing with a machine or another human (Turing, 1950). Seventy years later, with the advent of ChatGPT, Turing's formulation is still insightful and now increasingly relevant.

4 Computers and Communications

Computers would only be slightly more exciting than today's calculators if they were only capable of mathematical calculations. But it quickly became apparent that computers can exchange information and coordinate with other computers, allowing them (and people) to communicate and collaborate as well as compute. The far-reaching consequences of computing, the focus of this book, are due as much to computers' ability to communicate as to compute, although the latter attribute is more closely identified with the field.

Among the most ambitious early applications of computers were collections of devices and computers linked through the telephone system. SAGE, deployed in 1957, was a computer-controlled early warning system for missile attacks on the United States (na, 2023b). In 1960, American Airlines deployed Sabre, the first online reservation and ticketing system, which accepted requests and printed tickets on terminals worldwide (Campbell-Kelly, Martin, 2004). The significance of both systems went far beyond their engineering and technical challenges. Both directly linked the real world—World War II and commercial transactions—to computers without significant human intermediation. People did not come to computers; computers came to people. Starting with systems like these, these machines have increasingly intruded into everyday life.

Businesses using computers, e.g., American Airlines, quickly accumulated large quantities of data about their finances, operations, and customers. Their need to efficiently store and index this information led to the development of database systems and mass storage devices such as disk drives. Around this time, the implications of computers on people's privacy emerged as a general concern as the capacity of computers to collect and retrieve information rapidly increased. At that time, perhaps because of its traditional role, attention was focused more on government information collection than private industry (na, 2973).

Another fundamental innovation of that period was the ARPANET, the Internet's direct intellectual and practical predecessor. The US Department of Defense created the ARPANET in the late 1960 and early 1970 as a communication system that could survive a nuclear attack on the USA (Waldrop, 2001). The ARPANET's fundamental technical innovation was packet switching, which splits a message between two computers into smaller pieces that could be routed independently along multiple paths and resent if they did not reach their destination. Before, communication relied on a direct connection between computers (think of a telephone wire, the technology used at the time). These connections, called circuits, could not have grown to accommodate a worldwide network like today's Internet. Moreover, the engineering of the ARPANET was extraordinary. The network grew from a few hundred computers in the 1970s to tens of billions of computers today in a smooth evolution that maintained its overall structure and many of its communication protocols, even as new technologies, such as fiber optics and mobile phones, emerged to support or use the Internet (Mccauley et al., 2023).

5 Computing as a Science

In the 1960s and 1970s, the theory underlying computer science emerged as a discipline on its own that offered an increasingly nuanced perspective on what is practically computable. Three decades earlier, Turing hypothesized that stored program computers were universal computing devices capable of executing any algorithm—though not solving any problem, as he proved that no algorithm could decide whether any algorithm would terminate. Turing's research ignored the running time of a computation (its cost), which held no relevance to his impossibility results but was of first-order importance to solving real-world problems.

The study of these costs, the field of computational complexity, started in the 1960s to analyze the running time of algorithms to find more efficient solutions to problems. It quickly became obvious that many fundamental problems, for example, sorting a list of numbers, had many possible algorithms, some much quicker than others.

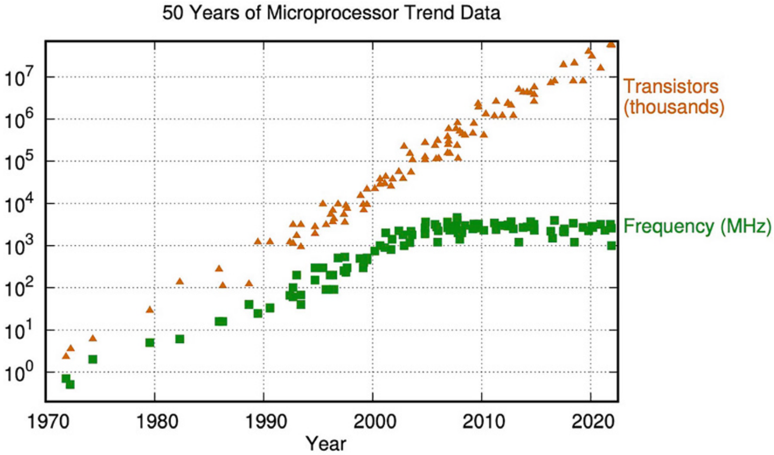
Theoreticians also realized that the problems themselves could be classified by the running cost of their best possible solution. Many problems were practically solvable by algorithms whose running time grew slowly with increasingly large amounts of data. Other problems had no algorithm other than exploring an exponential number of possible answers, and so could only be precisely solved for small instances. The first group of problems was called P (for polynomial time) and the second NP (nondeterministic polynomial time). For 50 years, whether P = NP has been a fundamental unanswered question in computer science (Fortnow, 2021). Although its outcome is still unknown, remarkable progress has been made in developing efficient algorithms for many problems in P and efficient, approximate algorithms for problems in NP.

Moreover, computer science's approach of considering computation as a formal and analyzable process influenced other fields of education and science through a movement called "computation thinking" (Wing, 2006). For centuries, scientific and technical accomplishments (and ordinary life—think food recipes) offered informal, natural language descriptions of how to accomplish a task. Computer science brought rigor and formalism to describing solutions as algorithms. Moreover, it recognized that not all solutions are equally good. Analyzing algorithms to understand their inherent costs is a major intellectual step forward with broad applicability beyond computers.

6 Hardware “Laws”

Computer science was extremely fortunate to ride on the back of an extraordinary and unprecedented improvement in silicon semiconductors, the underlying technology used to construct computers. The earliest computers were built from mechanical relays, which could switch on or off roughly 20 times per second. They were quickly succeeded by vacuum tubes, which could switch millions of times per second, but were large, hot, and unreliable. In the 1960s, transistors replaced tubes with much smaller, more reliable switches. More importantly, many transistors could be fabricated and wired together on a small piece of silicon called a “chip,” which offered compelling size, speed, and cost advantages. In 1965, Gordon Moore noted that the number of transistors on a chip doubled every year, an observation that came to be called “Moore’s law.” This geometric increase in capacity has continued for over four decades, albeit at a slower pace. A decade after Moore, Robert Dennard published rules for IC design, which quantified how the smaller, denser transistors resulting from Moore’s Law could run faster without consuming more power.

Figure 2 illustrates this remarkable progress. Moore’s law and Dennard scaling led to three decades of computers whose running speed doubled every other year, a remarkable period of innovation that ended around 2005, when electrical considerations made it impossible to continue running computers faster, even though the number of transistors on a chip continued to double. From the 1970s to the early 2000s, computers dropped rapidly in cost at the same time as their performance increased, which hastened the birth of the software industry (discussed below) and made possible increasingly ambitious uses of computers.



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

Fig. 2 Moore’s law and Dennard scaling. The number of transistors on a chip has doubled every other year for 50 years. For the first half of this period, each generation of chips also doubled in speed. That improvement ended around 2005. From Karl Rupp, CC BY 4.0

Another important observation, called Kryder's law, was that the amount of data that could be stored in a square centimeter also grew geometrically at a faster rate than Moore's law. This progress has also slowed as technology approaches physical limits. Still, storage cost fell from $82 million/Gigabyte (billion bytes) for the first disk drive in 1957 to 2 cents in 2018 (both in 2018 prices). This amazing improvement not only made richer and more voluminous media such as photos and video affordable, but it also made possible the collection and retention of unprecedented amounts of data on individuals.

7 Personal Computers

In the mid-to-late 1970s, the increasing density of integrated circuits made it possible to put a "computer on a chip" by fabricating the entire processing component on a single piece of silicon (memory and connections to the outside world required many other chips). This accomplishment rapidly changed the computer from an expensive, difficult-to-construct piece of business machinery into a small, inexpensive commodity that entrepreneurs could exploit to build innovative products. These computers, named microprocessors, initially replaced inflexible mechanical or electric mechanisms in many machines. As programmable controllers, computers were capable of nuanced responses and often were less expensive than the mechanisms they replaced.

More significantly, microprocessors made it economically practical to build a personal computer that was small and inexpensive enough that an employee or student could have one use to write and edit documents, exchange messages, run line-of-business software, play games, and do countless other activities.

With the rapidly increasing number of computers, software became a profitable, independent business, surpassing computer hardware in creativity and innovation. Before the microprocessor, software was the less profitable, weak sibling of hardware, which computer companies viewed as their product and revenue source. The dominant computer company IBM gave away software with its computers until the US government's antitrust lawsuit in the early 1970s forced it to "unbundle" its software from hardware. Bill Gates, a cofounder of Microsoft, was among the earliest to realize that commodity microprocessors dramatically shifted computing's value from the computers to the software that accomplished tasks. IBM accelerated this shift by building its iconic PC using commodity components (a processor from Intel and an operating system from Microsoft) and not preventing other companies from building "IBM-compatible" computers. Many companies sprung up to build PCs, providing consumer choice and driving down prices, which benefited the emerging software industry.

Moreover, the widespread adoption of powerful personal computers (doubling in performance every 2 years) created a technically literate segment of the population and laid the foundation for the next major turning point in technology, the Internet.

8 Natural Interfaces

Interaction with the early computers was textual. A program, the instructions directing a computer's operation, was written in a programming language, a highly restricted and regularized subset of English, and a computer was directed to run it using textual commands. Though small and precise, most people found these languages difficult to understand, limiting early machines' use. In the late 1960 and 1970s, graphical user interfaces (GUIs) were initially developed, most notably at Xerox PARC (Michael A. Hiltzik, 1999). They became widespread with the introduction of the Apple Macintosh computer in the early 1980s. These interfaces provided pictural metaphor-oriented interfaces directly manipulated through a mouse. This user interface change made computers accessible and useful to many more people.

The graph aspect of GUIs enabled computers to display and manipulate images, though initially, software treated them as collections of pixels and could not discern or recognize their content. This capability only came later, with the advent of powerful machine-learning techniques that enabled computers to recognize entities in images. In addition, the early computers were severely constrained in computing power and storage capacity, which limited the use of images and video, which is far larger than a single image.

Computers also adopted other human mechanisms. Voice recognition and speech generation are long-established techniques for interaction. Recently, machine learning has greatly improved the generality and accuracy of human-like speech and dialog, so it is not unusual to command a smartphone or other device by speaking to it.

Most computers do not exist as autonomous, self-contained entities, like PCs or smartphones with their own user interface. They are instead incorporated into another device and interact through its features and functionality. Mark Weiser called this "ubiquitous computing" (Weiser, 1991), where computing fades into the background, so no one is aware of its presence. Many of these computers, however, are accessible through the Internet, raising vast maintenance, security, and privacy challenges.

9 The Internet

The Internet started as a US government research project and infrastructure in the 1970s. Access was initially limited to the military, universities, and a few government-related businesses. In the early 1990s, two important events occurred. The US government agency managing the public Internet, the National Science Foundation (NSF), decided it was time to transition from a government-led project into a commercial product. In a little-heralded but enormously successful effort, it turned the Internet over to the technical community that built it and the private companies that operate the individual networks that comprise today's Internet.

The other crucial change was the emergence of the World Wide Web (the "Web") as the Internet's "killer app," which caused it to gain vast public interest and financial investment. While working at CERN, a physics research lab in Switzerland, Tim Berners-Lee developed a networked hypertext system he optimistically called the "World Wide Web (WWW)." CERN released his design and software to the public in 1991. A few years later, the University of Illinois's Mosaic browser made Berners-Lee's invention easier to use and more visually appealing on many types of computers. The academic community, already familiar with the Internet, rapidly jumped on the Web. Then, remarkably, both inventions made a rare leap into the public eye and widespread adoption. In a remarkably short time, businesses started creating websites, and the general population started to buy personal computers to gain access to "cyberspace."

Other chapters of this book discuss a remarkable spectrum of societal and personal changes in the past three decades. Underlying *all* of them are the Internet and the Web, which made it possible to find information, conduct commerce, and communicate everywhere at nearly zero cost. Before these inventions, there were two ways to communicate.

First, you could speak to another person. If the person was distant, you used a telephone or radio. However, both alternatives were expensive, particularly as distance increased, because the technical structure of telephone systems allocated a resource (called a circuit) to each communication and charged money to use it throughout the conversation. By contrast, the Internet used packet switching, which only consumed resources when data was transferred, dramatically lowering costs. In fact, users pay a flat rate in most parts of the Internet, independent of their usage, because finer-grained billing is neither necessary nor practical. In addition, for historical reasons, telephone companies were regulated as "natural" monopolies, which allowed them to keep their prices high. The Internet, in reaction, sought multiple connections between parties and resisted centralization and monopolization.

The second alternative, of course, was to engrave, write, or print a message on a stone tablet or piece of paper and physically convey the object to the recipient, incurring substantial costs for the materials, printing, and delivery. Moreover, paper has a low information density, requiring considerable volume to store large amounts of data. In addition, finding information stored on paper, even if well organized, takes time and physical effort.

Computing and the Internet *completely* changed all of this. A message, even a large one, can be delivered nearly instantaneously (at no cost). And data, stored electronically at rapidly decreasing cost, can be quickly retrieved. This is the *dematerialization of information*, which no longer needs a physical presence to be saved, shared, and used. This change, as much as any, is behind the "creative destruction" of existing industries such as newspapers, magazines, classified advertising, postal mail, and others that conveyed information in a tangible, physical form.

Another fundamental computer science innovation is public key cryptography, which made private communication and safe online commerce possible, enabling businesses to communicate and interact through cyberspace rather than the physical world. Cryptography hides the contents of messages so that only the sender and receiver can read them, even if they traverse the public Internet. And cryptographic protocols added functionality such as making it possible for two parties to identify each other online and authenticate a transaction.

10 Mobile Computing

The next important and radical change was mobile computing, which became practical when computers became sufficiently power-efficient (another consequence of Moore's law) to be packaged as smartphones. The defining moment for mobile computing was Apple's introduction of the iPhone in 2007 (Isaacson, 2011). It combined in a pocket-sized package, a touchscreen interface appropriate for a small device without a keyboard or mouse, and continuous connectivity through the wireless telephone network. For most of the world's population, smartphones are *the* access point to the Internet and computing. "Personal" computers never shrank smaller than a notebook and remained better suited to an office than as a constant companion. In less than a decade, the smartphone became an object that most people always carry.

Smartphones also changed the nature of computing by attaching cameras and GPS receivers to computers. Smartphone cameras dramatically increased the number of photos and videos created and let everyone be a photographer and videographer. They also exploited the vast computational power of smartphones to improve the quality of photos and videos to a level comparable with much larger and optically sophisticated cameras operated by skilled photographers. Their GPSs introduced location as an input to computation by continuously tracking a phone's location in the physical world. Location, like many features, is a two-edged sword that offers sophisticated maps and navigation and enables tracking of people by advertisers and malefactors.

Perhaps the most far-reaching consequence of smartphones is that they "democratized" computing in a form whose low cost and remarkably new functionality was quickly adopted by most people worldwide. Earlier computers were concentrated in the developed world, but smartphones are ubiquitous, with a high adoption even in less developed countries. The deployment of wireless networks in these countries brought the citizens of these countries to a nearly equal footing in terms of information access and communications.

11 Machine Learning

A recent and extremely significant advance in computing is machine learning (ML), the process of automatically inferring features in data collections and applying this inference to make predictions from and take actions on other, unseen data. For example, an ML system can be trained on a large collection of labeled photographs, e.g., a bird's photo might be labeled "bird, Herring Gull Larus." An ML model trained on a large collection of photographs could then analyze other photos, even of bird species not included in the training set, and recognize that the images contain birds. Several years ago, ML systems reached a human performance level in this computer vision classification task (Shankar et al., 2021).

Beyond image recognition, ML has been trained to mimic many human skills, such as computer vision, speech recognition, language translation, grammar correction, question answering, game playing, and others. In most cases, the key enabling factor is a large training set of labeled data. For example, large language models (LLMs) are often trained on tens or hundreds of billions of documents from the Web. OpenAI's ChatGPT was trained this way and can respond to general questions with articulate, well-formed responses and conduct a realistic dialog, albeit with many grievous lapses, reflecting a complete lack of understanding of the underlying meaning.

ML represents a fundamental change in how computers are programmed. For the first seven decades, programmers wrote explicit instructions to direct a computer to solve a task. ML shifted the perspective from "teaching" a computer to having the computer "learn" how to accomplish a task by observing past examples. This new approach has proven very successful in developing human-like skills for computers, which programmers found difficult or impossible to describe fully in a program. However, the shift leaves some people concerned that computers are becoming "intelligent" and might soon surpass human abilities (Bostrom, 2014).

These topics are discussed in more detail in the chapter by Woltran and Heitzinger in this volume.

12 Big Data and Cloud Computing

Underlying these advances in machine learning, and many other fields, is the ability to collect and analyze vast amounts of data, known as "Big Data." The hardware and software infrastructure for storing and processing this data was originally developed for Web applications such as search engines, which harness warehouses full of tens of thousands of computers to index most Internet pages and rapidly respond to user queries (Barroso et al., 2013). Each search triggers coordinated activity across thousands of computers, a challenging form of computation called parallel computing.

Internet search was made possible by advances in many fields of computer science, including computer design, high-bandwidth networking, inexpensive storage devices, and research on using multiple computers to solve a single task. The infrastructure was originally the proprietary asset of a few companies. Amazon democratized this form of computing with a product that came to be called “cloud computing.” It comprises low-cost computer access in Internet-connected data centers (the cloud) and sophisticated software for building reliable and scalable systems on collections of computers. Before this, a company needed to buy and manage its own computers, which limited the tasks that most companies could accomplish with their limited capital and expertise.

Cloud computing effectively removed the barrier to constructing large-scale computing systems. This has made it possible to collect, store, and analyze vast amounts of data, now a routine business practice. Many websites record every user interaction in detail, and these records are retained to provide the raw material to train machine-learned systems. This practice has serious privacy implications but is routine because data [aka the “new oil” (na, 2017)] promises to be the raw material to build profitable new businesses.

More benignly, the ability to collect and analyze large amounts of data is changing how other fields of science and engineering conduct research. Jim Gray called Big Data the fourth paradigm of scientific discovery (after observation, theory, and modeling) (Hey et al., 2009).

13 Security and Privacy

Because computers contain valuable information and control important devices and activities, they have long been the target of malicious and criminal attempts to steal data or disable their functions. The Internet greatly worsened these problems by making nearly every computer accessible worldwide.

Computer science has failed to develop a software engineering discipline that enables us to construct robust software and systems. Every nontrivial program (with a handful of exceptions) contains software defects (“bugs”), some of which would allow an attacker to gain access to a computer system. The arms race between the attackers and developers is very one-sided since an attacker only needs to find one usable flaw, but the developer must eliminate all flaws. Like security in general, mitigations—updating software to fix bugs, watching for attacks, and encrypting information—are essential.

Privacy is typically grouped with security because the two fields are closely related. Privacy entails personal control of your information: what you do, what you say, where you go, whom you meet, etc. However, privacy differs in a crucial aspect from security since the owners and designers of systems abuse privacy because this personal information has significant value that can be exploited. See the chapter by Weippl and Sebastian in this volume.

14 Conclusions

A natural question is whether computing's rapid growth and evolution can continue. As Niels Bohr said, "Prediction is very difficult, especially about the future." I believe computing will continue to grow and evolve, albeit in different but still exciting directions. New techniques to perform computation, for example, based on biology or quantum phenomena, may provide solutions to problems that are intractable today. At the same time, new inventions and improved engineering will continue to advance general-purpose computing. However, the enjoyable decades of exponential improvement are certainly finished. Computing will become similar to other fields in which improvement is slow and continuous.

The separate questions of whether computing's rapid growth was good or bad and whether its likely demise is good or bad can be evaluated in the context of the rest of this book. In many ways, this question is like asking whether the printing press was good or bad. Its introduction allowed the widespread printing of vernacular bibles, which supported the Protestant Reformation and led to decades of religious and civil war. Was that too large a cost to spread literacy beyond a few monks and royalty? Computing has also disrupted our world and will likely continue to do so. But these disruptions must be balanced against the many ways it has improved our life and brought knowledge and communication to the world's entire population.

Discussion Questions for Students and Their Teachers

1. Computers have grown cheaper, smaller, faster, and more ubiquitous. As such, they have become more embedded throughout our daily life, making it possible to collect vast amounts of information on our activities and interests. What apps or services would you stop using to regain privacy and autonomy? Do you see any alternatives to these apps and services?
2. Many aspects of computing work better at a large scale. For instance, an Internet search engine needs to index the full Web to be useful, and machine learning needs large data sets and expensive training to get good accuracy. Once these enormous startup costs are paid, it is relatively inexpensive to service another customer. What are the consequences of this scale for business and international competition?
3. Moore's law is coming to an end soon, and without new technological developments, the number of transistors on a chip will increase slowly, if at all. What are the consequences of this change for the tech industry and society in general?
4. Climate change is an existential threat to humanity. Because of their ubiquity and large power consumption, computers are sometimes seen as a major contributor to this challenge. On the other hand, our understanding of climate change comes from computer modeling, and computers can replace less efficient alternatives, such as using a videoconference instead of travel. What is the actual contribution of computing to global warming, and what can be done about it?

Learning Resources for Students

Many technical books and research papers describe the technical innovations mentioned above in great detail. They can easily be found with a search engine.

1. Lewis, H.R. (Ed.), 2021. *Ideas that Created the Future: Classic Papers of Computer Science*. MIT Press, Cambridge, MA.

   For convenience, many classic papers are collected in the volume edited by Harry Lewis.
2. Barroso, L.A., Clidaras, J., Hölzle, U., 2013. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 2nd ed, Synthesis Lectures on Computer Architecture. Morgan & Claypool.

   However, Lewis's collection misses recent papers and those concerned with practical aspects, such as building Internet-scale computer systems, which Barroso et al. cover well.
3. In addition, the tech field is caught in the public eye, and many excellent, accessible books talk about its history and technical aspects.
4. Waldrop, M.M., 2001. *The Dream Machine: J. C. R. Licklider and the Revolution that Made Computing Personal*. Viking.

   Waldrop's book on J.C.R. Licklider is a biography of the remarkable psychologist who led the development of interactive computing and the Internet at ARPA.
5. Hiltzik, M.A., 1999. *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age*. Harper-Collins.

   Hiltzik's book followed these ideas as they were incubated at Xerox PARC, a remarkable industrial lab.
6. Isaacson, W., 2011. *Steve Jobs*. Simon & Schuster.

   Isaacson's biography of Steve Jobs provides the other half of the story by showing how he made these ideas into two products that changed the world, the Apple Mac and iPhone.
7. Gleick, J., 2021. *The Information*. Vintage.

   Gleick's book dives into communication and information theory, the opposite side of the computational coin.

References

Barroso, L. A., Clidaras, J., & Hölzle, U. (2013). *The datacenter as a computer: An introduction to the design of warehouse-scale machines* (Synthesis Lectures on Computer Architecture) (2nd ed.). Morgan & Claypool.

Bostrom, N. (2014). *Superintelligence*. Oxford University Press.

Campbell-Kelly, M. (2004). *From airline reservations to sonic the Hedgehog*. MIT Press.

Fortnow, L. (2021). Fifty years of P vs. NP and the possibility of the impossible. *Communications of the ACM, 65*, 76–85. https://doi.org/10.1145/3460351

Gleick, J. (2021). *The information*. Vintage.

Hey, T., Tansley, S., Tolle, K., & Gray, J. (Eds.). (2009). *The fourth paradigm: Data-intensive scientific discovery*. Microsoft Research.

Hiltzik, M. A. (1999). *Dealers of lightning: Xerox PARC and the dawn of the computer age*. Harper-Collins.

Isaacson, W. (2011). *Steve Jobs*. Simon & Schuster.

Lewis, H. R. (Ed.). (2021). *Ideas that created the future: Classic papers of computer science*. MIT Press.

Mccauley, J., Shenker, S., & Varghese, G. (2023). Extracting the essential simplicity of the Internet. *Communications of the ACM, 66*, 64–74. https://doi.org/10.1145/3547137

na. 2973. Records, Computers and the Rights of Citizens (No. DHEW NO. 73-94). US Department of Health, Education, and Welfare.

na. (2023a). *IBM 701*. Wikipedia.

na. (2023b). *Semi-automatic ground environment*. Wikipedia.

na. (2017). *The world's most valuable resource*. The Economist.

na. (2003). IBM Archives: 701 Feeds and speeds [WWW Document]. Accessed 17 February 17, 2023, from www-apache-app.cwm.gtm.ibm.net/ibm/history/exhibits/701/701_feeds.html

Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., & Schmidt, L. (2021). Evaluating machine accuracy on ImageNet. *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind, LIX*, 433–460. https://doi.org/10.1093/mind/LIX.236.433

Turing, A. M. (1937). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, s2-42*, 230–265. https://doi.org/10.1112/plms/s2-42.1.230

Waldrop, M. M. (2001). *The dream machine: J. C. R. Licklider and the revolution that made computing personal*. Viking.

Weiser, M. (1991). The computer for the 21st century. *Scientific American, 265*, 94–105.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*, 33–35. https://doi.org/10.1145/1118178.1118215