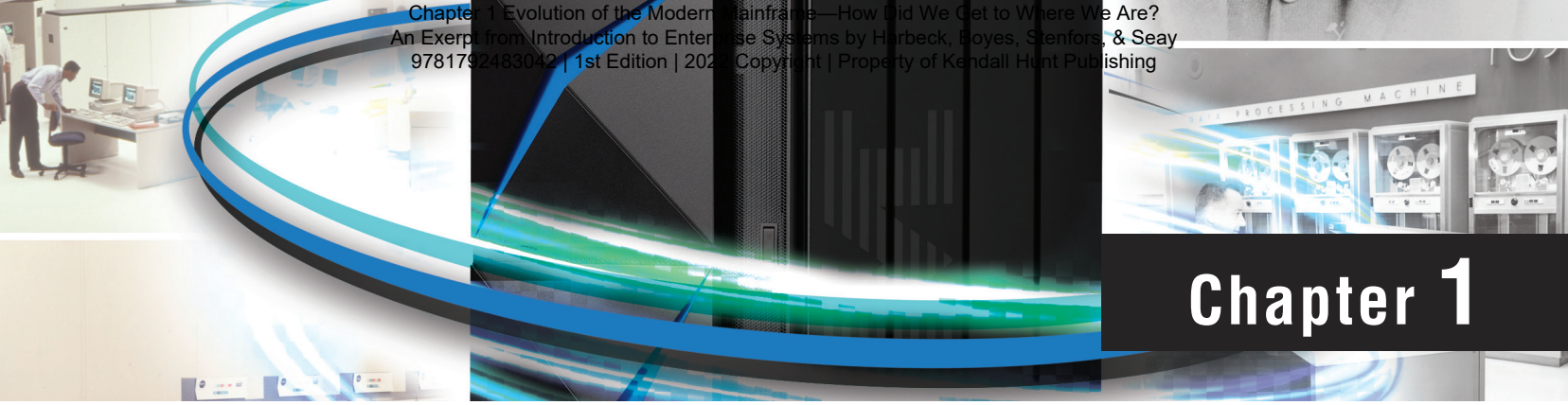
Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Excerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing



Chapter 1

Evolution of the Modern Mainframe—How Did We Get to Where We Are?

KH ALL RIGHTS RESERVED

In this section, we will cover:

- The origins of computation and the evolution of the digital computer
- The emergence of data processing as a specific discipline
- The thought leaders that lead to the creation of the IBM 360 platform
- Why this technology is important to the global economy

At the end of this section, you will have:

- An understanding of what the mainframe is and what it does
- An understanding of the history of computation

Introduction

Throughout human history, people have sought to make information persistent and functional external to their own personal recollections. This has led to innovations ranging from written language to monetary objects such as coins to counting devices such as the abacus and record-keeping methods such as ledgers. In each of these cases, the nature of the innovation was to serve human and business purposes.

While numbers and non-numeric data grew up in parallel, and were often interspersed in the same documents, the rules for handling each were different. By themselves, numbers could be used for calculations, and were often seen as a type of universal language, particularly once the Hindu–Arabic system became the norm. Data that included non-numeric content tended to be more context-sensitive, including the type (narrative, song, legal, etc.) and language. However, it was subject to one of the greatest historical automations: the Gutenberg printing press.

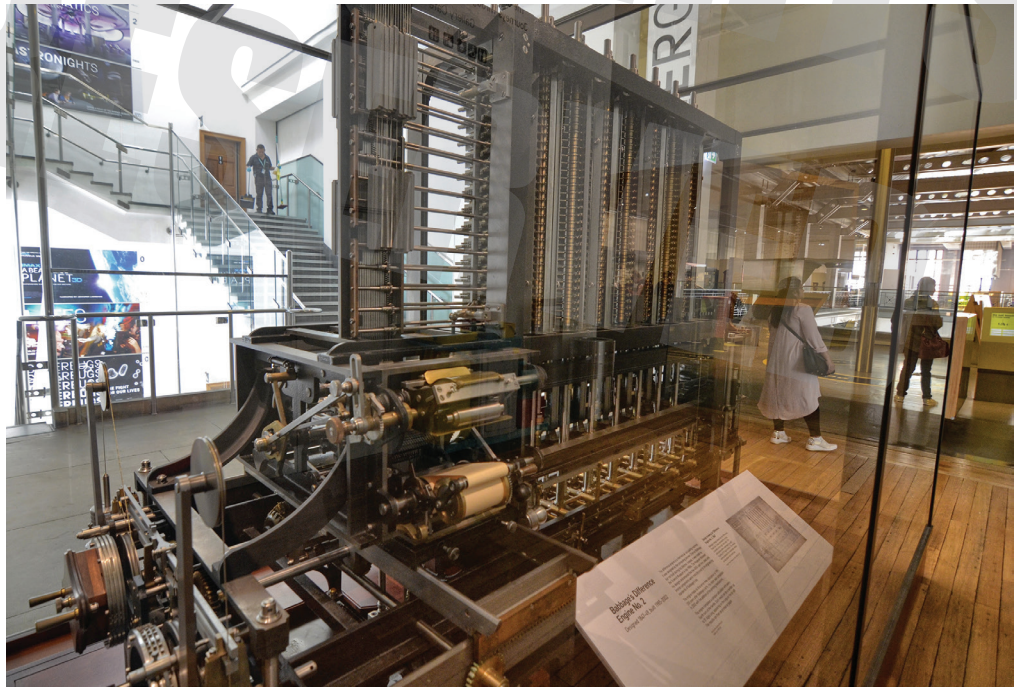
The Industrial Revolution heralded the invention of mechanized automation, significant advancement in calculation, and early examples of data processing. But it was punched cards that were to bring this all together—beginning with Jacquard's loom, which used punched card data to embody a repeatable program for creating tapestries, at the

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

beginning of the nineteenth century. Punched cards then became a key component of Charles Babbage's proposed Analytical Engine, leading Lady Ada Lovelace to propose the first computer program, describing how Bernoulli numbers could be calculated.

© Lesley Photograph/Shutterstock.com

Abacus



© Massimo Parisi/Shutterstock.com

Babbage's Difference Engine

All Rights Reserved

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

By the dawn of the twentieth century, mechanical adding machines and cash registers were commonly used for processing numeric data—often monetary—and punched cards were beginning to enter broad usage for recording numeric data. However, non-numeric data was only being automated to the extent of advances in printing presses and the invention of manual typewriters.

It wasn't until after the end of World War II, when serious efforts had begun to develop electronic computing technology, that numeric and non-numeric data were brought together and treated equivalently as data to be processed, just four years after the concept of storing programs and their data in the same place was developed.[1] Specifically, a Jesuit priest named Fr. Roberto Busa met with IBM CEO Thomas J. Watson Sr. in November of 1949 and convinced him to make non-numeric characters available as data on IBM's punched card machines, so he could use them for an index of the works of Thomas Aquinas that he was trying to create.

The merging of data types made it possible to write computer programs using non-numeric characters, leading Dr. Grace Hopper (later Rear Admiral) in 1952 to suggest the idea of a compiled language, written in human-friendly phrases and then compiled into the numeric language that computers actually speak.

Throughout the subsequent decade, as computing hardware advanced iteratively with input from users, more and more principles were discovered and refined for quality computing. A major leap forward in this process was the 1955 founding of the world's first computer user group, SHARE, which provided active input to IBM as they continued to develop their products. By 1959, insights such as Hopper's were leading to the creation of the COBOL compiled programming language, which embodied many of the lessons learned about how to process data effectively using computing technology. Those same lessons were concurrently being learned and applied by IBM in their continuing advancement of their computing hardware, leading to their ultimate computing platform.

As the 1960s began, it became clear to the people at IBM that they had to move beyond the current level of computing technology they were creating and take a leap forward, creating a "SPREAD" committee to develop a brand new generation:

> The SPREAD committee produced its report at the end of 1961, and the Corporate Management Committee adopted its recommended New Product Line as the successor for all existing product lines. This stunningly bold move was later called, by Fortune magazine, "IBM's $5,000,000,000 gamble." Evans called it "You bet your company."[2]

---

[1] https://web.archive.org/web/20130314123032/http://qss.stanford.edu/~godfrey/vonNeumann/vnedvac.pdf; http://none.cs.umass.edu/~dganesan/courses/fall09/handouts/Chapter4.pdf

[2] Frederick P. Jr. Brooks, *The Design of Design: Essays from a Computer Scientist* (Boston, MA: Pearson Education, 2010), Kindle Edition location 4810.

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

Consequently, on April 7, 1964, IBM announced its ultimate computing platform, the System/360 mainframe. Programs that were written for any of the six different models would run on all of them, and would continue to run on future upgrades to this platform. As a consequence, constant rewriting of programs ceased to be necessary every time a new computer model came out. Indeed, there are likely programs still in use on modern IBM Z mainframes that were originally written for System/360.

Of course, the journey didn't end there. While the design and architecture of the platform were carefully researched, and embodied all the lessons learned about general-purpose computing to that point, there were many new lessons that could only be learned now that the stability of a consistent platform had been achieved. These included multi-user systems, databases, security, virtualization, and a wide range of other technologies and practices that grew up on the IBM System/360 mainframe and its successors, and which often continue to have their highest manifestation in that context.

Adoption of this new platform was so pervasive that IBM had to open it up to explicit competition in the 1970s by charging for individual software, hardware, and services offerings, rather than making the mainframe available for a single all-inclusive price.

Early successes included supporting the Apollo moon missions, but more typical was the way the IBM mainframe became essential to the processing of much of the key business data around the world, a role it retained even through the arrival of consumer computing platforms in the 1980s, and continues to have today.

An endless flood of relevant innovations has been part of the journey of mainframe hardware, operating systems, other software, and other aspects of the ecosystem, for over half a century. Some of these have been copied by smaller platforms, and some have been adopted from those platforms. An important example of the latter is Linux, which made its debut on the IBM mainframe in 1999.

Today's IBM Z mainframe continues to be the standard bearer for world-class quality, security, capacity, reliability, and functionality. Now, let's dig into some of the details about how we got here.

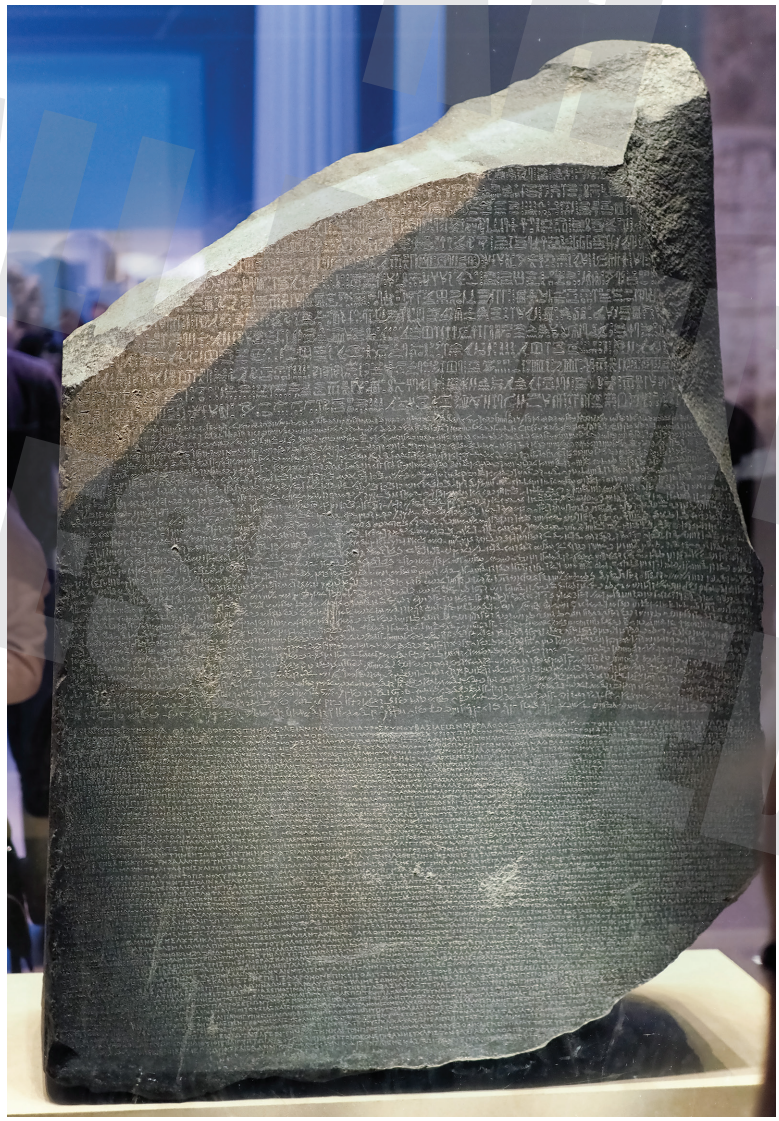
Pre-Electronic Devices Before the Industrial Revolution

Did humanity invent language or did language invent humanity? This chicken-and-egg question is more rhetorical than resolvable (or you may wish to characterize it as a "religious question"), but it does point out that the concept of distilling reality into data and storing, processing, and sharing it is of the essence of the human journey, and has been pervasive in everything we call progress.

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

Devices and methods for counting, tabulating, and recording have existed since the dawn of history, and while their use was not sufficiently pervasive to displace human abilities such as calculation and recollection, the ability to reliably retain, transmit, or produce consistent results made them indispensable parts of history as we have come to know it, and just like language, so these devices have likely shaped our thinking about processing numbers and other data. Most of us are familiar with the abacus, and also likely familiar with stone tablets that retain messages for millennia—such as the Rosetta Stone, which has a public message in three different scripts and two different languages.



© Claudio Divizia/Shutterstock.com

Rosetta Stone

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

Generally, art and craft were essential to the creation of objects that stored textual data—from inscriptions on sculptures to scrolls and codices—and those that processed numeric data, such as the abacus. Early science and math played both motivating and innovating roles, particularly in the development of devices to measure volume, weight, length/circumference, and time.

While various methods for reproducing multiple copies of the same text or pictures, for example signet rings, had been around for a long time, it wasn't until the fifteenth century that the concept was mechanized and elaborated. Then, in 1439, the printing press with moveable type was invented by Johannes Gutenberg, an important step forward in automating the generation and processing of textual data.

© Irina Budanova/Shutterstock.com

Gutenberg Press

ALL RIGHTS RESTRICTED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

It took another two centuries for mechanical devices to do more advanced calculation than the abacus, beginning with Napier's Bones in 1617 and Pascal's mechanical calculator, or Pascaline, in 1645. While representing Roman or Hindu–Arabic numbers in reproducible texts was one level of automation, creating devices for interacting with the rules specific to numbers was a much higher level achievement. Once it was clearly illustrated that machines could perform such tasks, formerly believed to be the exclusive province of human intelligence, the stage was set for an unlimited journey forward, just on time to usher in the looming Industrial Revolution beginning just over a century later, around 1760.

Steam Punk Data Processing: The Industrial Revolution

Indeed, looming was one of the key aspects of the industrial revolution, as the creation of textiles was automated, along with other mechanical tasks, using power sources that predated the general availability of electricity.

The modern term "steampunk" often refers to machines and similar art made with the metallic robustness we tend to associate with this era. And while coal and steam were key drivers of the energy for these mechanisms, their designs generally included elaborations on the simple machines described by the classical Greeks and Romans, and prefigured the way many modern mechanical devices still function.

It is important not to accidentally glorify this era in history, as many poor people, young and old, were sacrificed in the creation of the goods these machines were designed to deliver, through overwork, hazardous conditions, and other deprivation. Nonetheless, it is our history, and legacy, that we conflated the worth of humanity and technology, as illustrated so well in Mary Shelley's 1818 novel ***Frankenstein; or, The Modern Prometheus***.

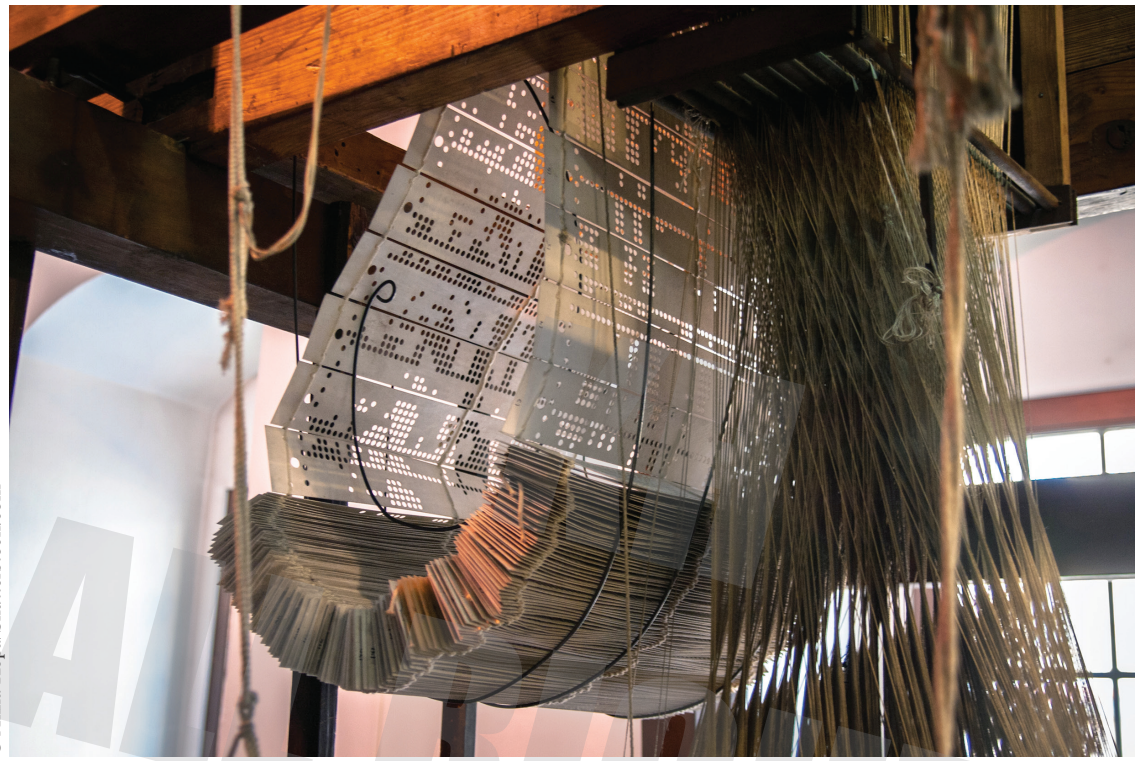
Concurrent with this conflation was the emergence of machine behaviors traditionally associated with human intelligence, especially at the peak of the Industrial Revolution and subsequently in the nineteenth century. One of the most important of these was automation driven by punched cardboard cards, as manifested in the Jacquard Loom, invented by Joseph Marie Jacquard in 1804. An unlimited number of complex patterns could be programmed using these cards, fed into the machine, and bring about a reproducible complex result—the most famous of which was a self-portrait of Jacquard himself.

The concept of these cards was subsequently applied by Charles Babbage in his plans for two early computing machines: the Difference engine (1822) and the Analytical engine (1837). These mechanical computing devices, which would have been powered by steam, were never actually constructed during his lifetime. They were designed to involve punched cards in their processing, an idea which inspired Lady Ada Lovelace to write a number of programs that would run on the Analytical engine.[3]

[3] Waldrop, M. Mitchell, *The Dream Machine* (Stripe Press, 2018, Kindle Edition).

All Rights Reserved

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing



© Anna Kepa/Shutterstock.com

Jacquard Loom

The Dawn of Modern Computing, Tabulating, and Recording

As the Industrial Revolution reached its fulfillment, the era of utility electricity began, and new inventions, while often still mechanical, increasingly involved electricity when power was needed.

So, typewriters (beginning in 1874) and cash registers (beginning in 1879) were able to manually enable the processing of textual and numeric data in response to direct human input. But some types of data required more processing than manual efforts would allow.

Chief among these was census data. Specifically in the United States of America (US), there was a constitutional requirement to do a census every decade, and the time it took to manually process the resultant data was rapidly approaching a decade, when the US government requested proposals for an automated way to process that data in a more manageable amount of time.

This request was based, of course, on the belief that such a means could actually exist, and this was partly based on input from Herman Hollerith, who had previously worked at the Census Bureau, and had also investigated the use of electromechanical means

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

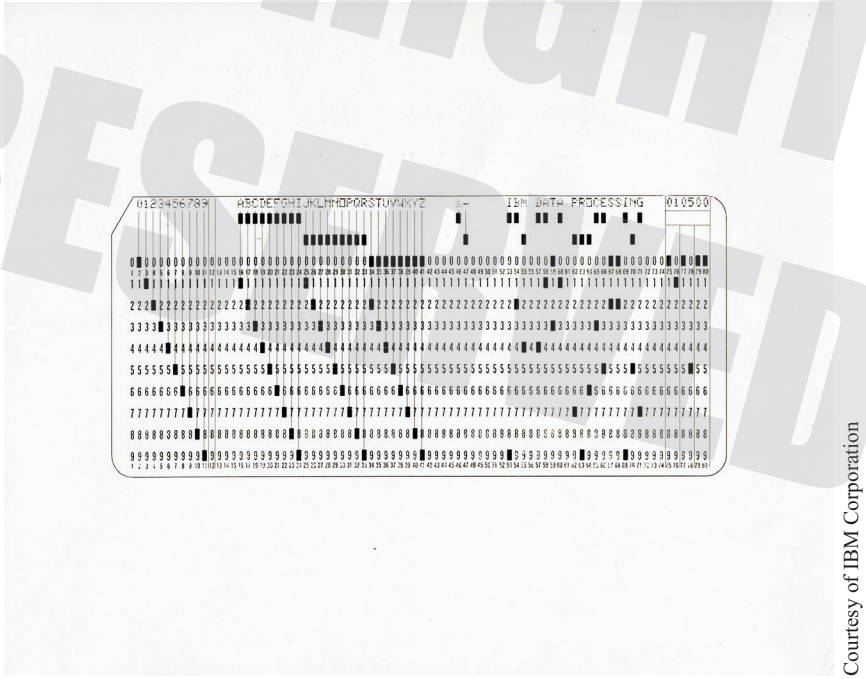
for tabulating data. So, of course, when the request went out, Hollerith was one of the respondents, and as it turned out, the successful bidder.

As a result, Hollerith created a punch-card-driven tabulating machine for census data, which worked so well that he was subsequently able to sell it to other governments around the world for the same purposes.

Because it was about tabulation, the information stored on these cards was effectively all numeric, and that would remain the only kind of information generally used on punch cards until 1949.

Now, Hollerith was a good inventor, but not a skilled business person, so he ended up selling his company into a merger with two other companies that had greater prospects for success as a merged entity. So it was, in 1911, that the Computing-Tabulating-Recording company was born, and the role of punch cards was historically established. Within thirteen years it was renamed, under its president Thomas J. Watson Sr., to International Business Machines, or IBM.

Then, in 1928, IBM introduced the 80-column punch card system[4], and the pattern was set. While some further refinements to form occurred during the subsequent years, and while other manufacturers offered cards with different capacities, the standard of eighty columns became a legacy that is still reflected today in the standard format of



Courtesy of IBM Corporation

IBM 80-column punch card

[4] Emerson W. Pugh, *Building IBM (History of Computing)* (p. 48) (The MIT Press, 1995), Kindle Edition.

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

text storage on the IBM mainframe, where the default width for text records is still eighty characters wide.

Electromechanical Computers, Computing Theory

If tabulating census results was the necessity that begat Hollerith's punch card machines, breaking enemy encryption may well have been a progenitor of modern electronic computers during World War II.

Anyone who has seen the movie The Imitation Game will likely recall Benedict Cumberbatch's portrayal of Alan Turing, inventor of an electromechanical "bombe"[5] computing device in Bletchley Park (about 50 miles northwest of London) that was used to solve the German Enigma code. Concurrent with this device was the more famous Colossus electronic computer, also used for cryptographic code analysis during the war.

Turing's ability to construct such an instrument was foreshadowed by computing theory he had already developed before the war, particularly including his 1936 Turing machine,[6] a theoretical computer that had the essential philosophical properties of a generic computer, and was able to demonstrate a wide range of assertions about the nature of computing which still hold true today.

Meanwhile, theory and practice continued to advance in North America as well, as experts such as mathematician John von Neumann[7] worked together on the Manhattan Project to develop the atomic bomb. This led to von Neumann's involvement with the ENIAC computer and his developing and elaborating the stored program concept, which came to be known as the "von Neumann Architecture" in 1945, and is the basis for how computer programs and their data are all stored together in computer memory.

One more initiative of significance during this time was the creation of the Harvard Mark I electromechanical computer.[8] It was a cooperative effort between IBM's

[5] Eric G. Swedin and David L. Ferro, *Computers: The Life Story of a Technology (Greenwood Technographies)* (Greenwood Publishing, 2005), A. Kindle Edition.

[6] Derek C. Schuurman, *Shaping a Digital World: Faith, Culture and Computer Technology* (InterVarsity Press, 2013), Kindle Edition.

[7] Eric G. Swedin and David L. Ferro, *Computers: The Life Story of a Technology (Greenwood Technographies)* (Greenwood Publishing, 2005), A. Kindle Edition.

[8] M. Mitchell Waldrop, *The Dream Machine* (Stripe Press, 2018), Kindle Edition.

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

© Giorgio Rossi/Shutterstock.com

German Enigma machine

Courtesy of IBM Corporation

Harvard Mark I

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

Clair D. Lake's team and Harvard's Howard H. Aiken.[9] It was also the last great non-electronic computer, and marked the parting of ways between IBM and Harvard University. Among the people who worked on this important technology was Dr. Grace Murray Hopper,[10] (later Rear Admiral), who was a pioneer of compilers.[11]

Electronic Computing

Early Electronic Computers

At the conclusion of the Second World War, many experts and technical and financial resources were freed from their wartime focus to support peacetime initiatives—as well as to advance the Allied cause in the Cold War. Likewise, IBM, which had been deeply invested in Allied success during the war, began various initiatives to advance digital computing. They also had many surplus punch card machines, no longer required by the military, which needed to be repurposed, and which played a formative role in the development of electronic computing—including their 80-column record length.

Building on Turing's and von Neumann's insights and theories, various types of storage, memory, and input and output devices were tried in addition to the advancing capabilities of processors. In the early days, data entry was accomplished using toggle switches, punched cards, and other primitive means, and the data were limited to numeric, which was consistent with the perspective of computers just being fancy calculators. But there were visionaries who believed in more.

One of the earliest visionaries was not a technologist at all. In fact, he was an Italian Jesuit priest with a PhD in Theology: Fr. Roberto Busa. He was on a quest to create a lemmatized index of the works of Thomas Aquinas (which he eventually did, and is available at https://www.corpusthomisticum.org/it/index.age). This was a massive undertaking, and would require some type of automation if it were to be completed in his lifetime, so he seized on the idea of using punch cards. Just one problem: available punch card machines were only capable of processing numeric data, and he needed to process textual data.

ALL RIGHTS RESERVED

[9] Emerson W. Pugh, *Building IBM (History of Computing)* (p. 75) (The MIT Press, 2005), Kindle Edition.

[10] M. Mitchell Waldrop, *The Dream Machine* (Stripe Press, 2018), Kindle Edition.

[11] Emerson W. Pugh, *Building IBM (History of Computing)* (p. 193) (The MIT Press, 2005), Kindle Edition.

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

So, Fr. Busa arranged to meet with IBM President Thomas J. Watson Sr. at their Manhattan office in November of 1949, where he made his case. Here is how he later described the encounter in his own words:

> I knew, the day I was to meet Thomas J. Watson, Sr., that he had on his desk a report which said that IBM machines could never do what I wanted. I had seen in the waiting room a small poster imprinted with the words: "The difficult we do right away; the impossible takes a little longer," [. . .] . I took it with me into Mr. Watson's office. Sitting in front of him and sensing the tremendous power of his mind, I was inspired to say: "It is not right to say 'no' before you have tried." I took out the poster and showed him his own slogan. He agreed that IBM would cooperate with my project until it was completed.[. . .] . That was providential![12]

The consequences of this meeting were historical. While it may be suggested that, if Busa hadn't succeeded, eventually someone else would have, the early timing of this event was critical to every subsequent innovation in computing, beginning with the concept of compilers.

Now that non-numeric characters were being used on punch cards, it was obvious that they would also be used in computers, so entering computer programs just as a sequence of numbers ceased to be unavoidable, especially in the mind of visionary Dr. Grace Hopper, who imagined telling computers what to do in English-like text rather than numbers. So the concept of the compiler was born, with advantages that included more intuitive programming, and portability, since a text-based source program could be compiled (and recompiled) for any computing architecture that had a compiler for the programming language the source was written in.

By 1953, a number of compiled languages had been created, including "A-2" which was a functional version of earlier efforts by Dr. Hopper and her colleagues, and was in use on the UNIVAC platform.[13] Seven years later, with involvement from Dr. Hopper, the COBOL (COmmon Business Oriented Language) programming language had been defined, based on all the lessons learned so far about how computing could serve business.

---

[12] Steven E. Jones, *Roberto Busa, S. J., and the Emergence of Humanities Computing (p. 32)* (Taylor and Francis, 2016), Kindle Edition.

[13] Emerson W. Pugh, (p. 194) (The MIT Press, 2005), Kindle Edition.

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

In addition to all of these innovations and insights, there was a dawning realization that computers were more than just fancy calculators. The stored program concept made it clear that arbitrarily complex programs could be written. Nonnumeric data opened up the idea of their being used for more than just math. And so the architectures began to be adapted to more and more general purposes throughout the 1950s.

In 1955, the SHARE user group—the world's first computer user group—was founded to share programs and insights for making better use of IBM's 701, 704, and subsequent computers as they grew into their roles of processing data in a wider and wider range of ways. In fact, in 1959, SHARE even made their own operating system available for the IBM 709 computer, known as SOS, an acronym for SHARE Operating System.[14]

As the 1960s dawned, many discoveries had been made and problems solved, both in hardware and software, and the ultimate manifestation of all of this was just around the corner.

The Characters of Computing

As we wade through all this history of technology, it's important to take a moment to point out that all of this has been by, for, and about real human beings. Electronic computers have been invented, designed, optimized, and adapted to meet the needs of people, whether those needs are business, academic, military, or any other aspect of our shared humanity. And it has been human beings of excellence, motivation, and vision who have made it happen.

Various histories of computing have made an effort to identify who the key contributors were—the endnotes and bibliography of this text offers some good examples. But, of course, for every person highlighted there were countless others who also contributed, and today we are all able to participate in the ongoing journey of developing and adapting computing to humanity's needs.

That said, more by way of illustration than in any sense exhaustive or complete, here are a few of the figures that have been mentioned up to this point, just to give some human provenance to this great technological journey. The reader is encouraged to search the Internet for more insights about each of these.

[14] Emerson W. Pugh, (p. 188) (The MIT Press,2005), Kindle Edition.

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

- Johannes Gutenberg (c. 1400–1468). Inventor of the moveable type printing press.

© Marzolino/Shutterstock.com

- Blaise Pascal (1623–1662). Theorist, technologist, inventor of a mechanical calculator. The Pascal programming language was named after him.

© Morphart Creation/Shutterstock.com

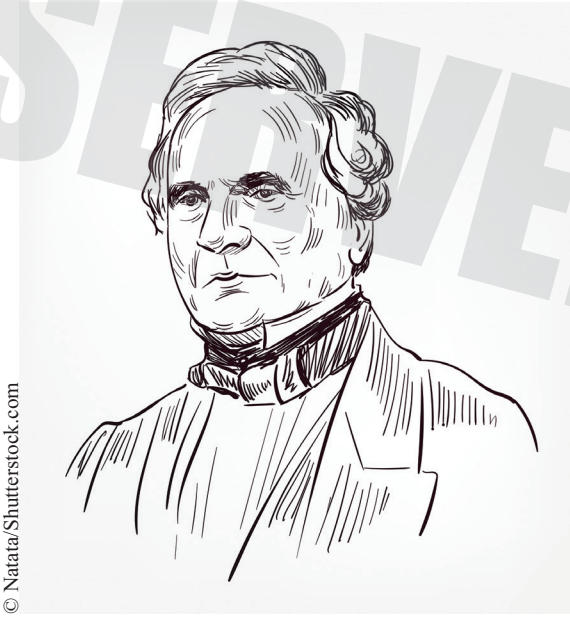
ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

- Joseph Marie Jacquard (1752–1834). Inventor of the punch-card-programmable loom.

© Boris15/Shutterstock.com

- Charles Babbage (1791–1871). Designer of the Difference engine and Analytical engine, including how punch cards would be used for such computing.



© Natata/Shutterstock.com

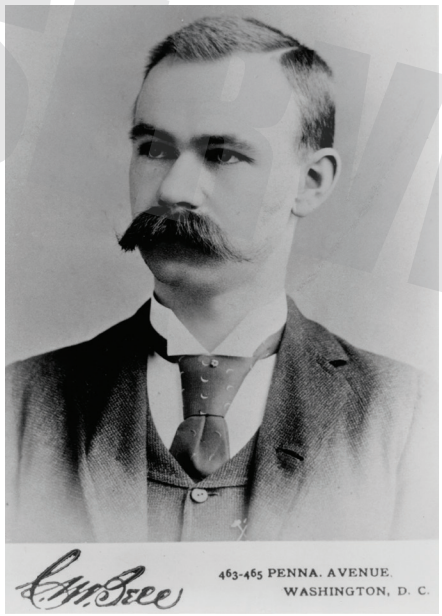
ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

- Augusta Ada (Byron) King, Countess of Lovelace (aka Lady Ada Lovelace, 1815–1852). Inventor of the concept of programming using punch cards, having written to Charles Babbage with a proposed program for his computing equipment to calculate Bernoulli numbers. The Ada programming language is named after her.

© Natata/Shutterstock.com

- Herman Hollerith (1860–1929). Inventor of the punch-card-tabulating machine used for census tabulation in the United States and other countries. Founder of one of the companies that merged into CTR in 1911 which became IBM in 1924.



Courtesy of IBM Corporation

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

- Thomas J. Watson Sr. (1874–1956). President of IBM, who renamed CTR to IBM, and played a pivotal role in the early development of electronic computing.

Courtesy of IBM Corporation

IBM's Thomas J. Watson, Jr., announcing System/360 with the compass rose logo in the background

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

- John von Neumann (1903–1957). Electronic computing pioneer, inventor of the stored program concept, or "von Neumann Architecture."

© catwalker/Shutterstock.com

- Rear Admiral Dr. Grace Brewster Murray Hopper (1906–1992). Computing and compiling pioneer, invented one of the first compilers, was involved with the success of COBOL.

Source: United States Navy

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

- Alan Mathison Turing (1912–1954). Originator of important computing concepts such as the Turing Machine and the Turing Test for Artificial Intelligence. Worked on important computing initiatives, including solving the German Enigma code in the Second World War.

© Natata/Shutterstock.com

- Fr. Roberto Busa (1913–2011). Responsible for advancing the cause of text processing, convincing IBM to make non-numeric characters available on their punch card machines. Led the creation of the Index Thomisticus.

Courtesy of IBM Corporation

MH ALL RIGHTS RESERVED

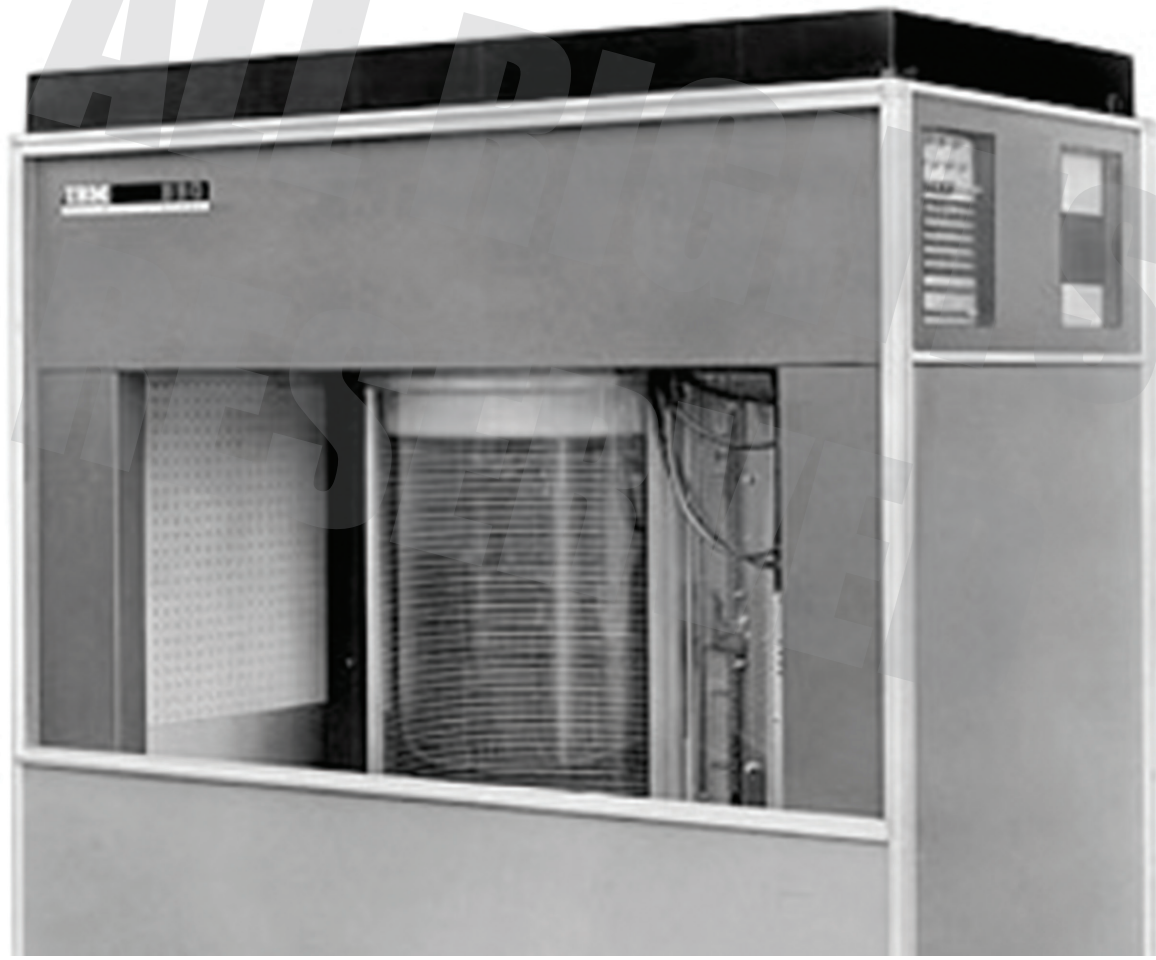
Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

Clearly, many thousands of other people could also be legitimately on this list, not to mention those who were already building their careers but whose most noteworthy contributions to electronic computing occurred after 1960.

The Cost of Resources

Ever since Gordon Moore made his discovery in 1965 that the capacity of integrated circuits doubled every year or so, people have taken for granted that computing resources would continually become less expensive and more plentiful given time. While this has ceased to be the case as of the writing of this textbook, the culture of distributed computing has tended to take it for granted.

However, the first electronic computers, up to and including the System/360 which was first announced in 1964 (see below), were the products of a more frugal mindset, by necessity.



Courtesy of IBM Corporation

The IBM 350 Disk Storage Unit

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

As the various raw materials and technologies were tried, sometimes kept, often eventually discarded, their costs and availability made it very expensive to create, and consequently employ, electronic computers.

Primary storage (aka memory) employed a wide range of media, from vacuum tubes to tiny ferrite magnetic rings or "cores" to integrated circuits. Secondary storage employed rotating drums and various kinds of disks, among other technologies. And their capacities were very limited: in 1956, the IBM 350 Disk Storage Unit had a capacity of 5 megabytes and rented for $650 per month.[15]

Peripheral devices were likewise costly and low capacity. In fact, IBM reduced the capacity of some devices that they received back from the military after the Second World War to rent them at lower prices.[16]

As a result, design decisions were not merely a function of technical possibility and customer desire, but had to take into account the price and availability of raw materials and the prospect of making a competitive result available to the customer at an acceptable price. So innovations were often focused on business value, and not merely technological capacity. Hence raw speed could be sacrificed for the capacity to produce a range of different results within desirable parameters of time and expense.

This also led to a design decision that is less typical in distributed computing, but remains intrinsic to the mainframe: protecting the central processor from being slowed down by interruptions from non-core activities, such as I/O, preferring instead to have secondary processors handle such technical activities and allow the central processor to focus on business results. This remains a hallmark of the mainframe (now called IBM Z) to the current day.

Production-Quality Early Computers

IBM is unquestionably a necessary player in the history of computing, and continues to be a leading light in all things computing-related. But they have never been the only player. Various academic institutions such as Harvard and Columbia, various military and government organizations, and business interests that participated as researchers, manufacturers, and customers were involved. And there was both competition and collegiality between IBM and these other organizations.

By the 1960s, IBM achieved such a leading role that their competition was grouped together as the alternatives to IBM, leading to the nickname of "Snow White and the

[15] Emerson W. Pugh, (p. 226), (The MIT Press, 2005), Kindle Edition.

[16] Kevin Maney, *The Maverick and His Machine: Thomas Watson, Sr. and the Making of IBM* (Kindle Locations 3170–3171, 2003), Kindle Edition.

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

Seven Dwarfs." Snow White was IBM. The Seven Dwarfs were Burroughs, Control Data, General Electric, Honeywell, NCR, RCA, and Sperry Rand.[17]

These various organizations produced a range of early computers that often were better at illustrating—or disproving—a concept rather than providing production value. But there were some computers that did make it to the market, or at least to public awareness, notably including the ENIAC in 1945, the world's first large-scale, programmed, electronic computer,[18] and the UNIVAC in 1949.[19]

IBM's early computing journey included a range of machines for different purposes, from scientific, to military, to decimal math for accounting-type purposes. In 1948, the IBM Selective Sequence Electronic Calculator (SSEC)[20] was, as its name suggested, a very fancy calculator. Likewise, the 604 electronic calculator was made available 1948.[21] In the 1950s, the 700-series computers became more and more advanced, beginning with the 701 "Defense Calculator" in 1952,[22] continuing through the 1953 announcement of the IBM 702 Electronic Data Processing Machine (EDPM),[23] the 704 and 705 EDM (Electronic Data Processing Machine) computers announced in 1954,[24] and finally the 709 in 1957. After that, the 7090 was delivered in 1959, along with the 1401 Data Processing System[25] and then the Stretch supercomputer (subsequently designated the IBM 7030) in 1961.[26] Each new model had new features and capacities and speeds, and many lessons were learned about what was needed from a computer made for specific purposes.

The Concept of General-Purpose Computing

A key aspect of the journey to modern electronic computing was defining the functionality and roles expected from computers. Starting as glorified calculators, and processing data for a wider and wider range of business, military, scientific, and other customers, the new breakthroughs didn't change the fact that the main discoveries and innovations tended to point in the same direction as Turing's original theoretical

[17] Emerson W. Pugh, (p. 296) (The MIT Press, 2005), Kindle Edition.
[18] Emerson W.Pugh, (p. 83) (The MIT Press, 2005), Kindle Edition.
[19] Emerson W. Pugh, (p. 139) (The MIT Press, 2005), Kindle Edition.
[20] Emerson W. Pugh, (p. 124). (The MIT Press, 2005), Kindle Edition.
[21] Emerson W. Pugh, (p. 168). (The MIT Press, 2005), Kindle Edition.
[22] Emerson W. Pugh, (p. 171) (The MIT Press, 2005), Kindle Edition.
[23] Emerson W. Pugh, (p. 175) (The MIT Press, 2005), Kindle Edition.
[24] Emerson W. Pugh, (p. 177) (The MIT Press, 2005), Kindle Edition.
[25] Emerson W. Pugh, (p. 266) (The MIT Press, 2005), Kindle Edition.
[26] Emerson W. Pugh, (p. 235) (The MIT Press, 2005), Kindle Edition.

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

machine: any true computer could process anything any other computer could process (though with differing qualities of service).

So, while a given configuration, memory capacity, speed, and set of peripherals might best suit a given workload, that level of configuration increasingly looked more appropriate as a consideration after the basic architecture of the platform was designed, rather than as input to customize that architecture.

Conceptually, this is similar to taking toy plastic building blocks that all fit the same connections and using them to build unique structures based on a generically consistent platform. And it allowed every different custom configuration to take advantage of the same leading edge of technology inherent in the most advanced possible base platform.

By focusing design and development efforts on a single leading edge platform, duplicate effort could be avoided, and shared benefit from discoveries and advances could be achieved for all kinds of customer, in addition to which, such a general-purpose platform would have the maximum potential to effectively host new workloads not foreseen by its designers.

System/360 and Successors

System/360

As will be detailed in following chapters, the lessons learned during the first two decades of electronic computing, including the utility of a general-purpose approach, coalesced into IBM's design, development, and delivery of the System/360 hardware and its OS/360 operating system, as described in Dr. Fred Brooks' 1975 landmark book, "The Mythical Man-Month: Essays on Software Engineering" as well as its 2010 companion volume, "The Design of Design: Essays from a Computer Scientist."

Brooks led the development of both, with very able involvement of a team of exceptional experts such as Gene Amdahl, but it was the decision by IBM President Thomas J. Watson, Jr., who succeeded his late father in 1956, to risk $5 Billion dollars in what was described as a "you bet your business" initiative that made it possible. Moving away from a different architecture for each different type of user such as scientific or business or military, a new general-purpose architecture would be designed that could be configured to meet the needs of any user.

The IBM's/360—The Mainframe

So it was, on April 7, 1964, that IBM announced the System/360. The 360 referred to the 360 degrees of the compass—addressing scientific as well as commercial computing. Its logo was a stylized "compass rose" which pointed to every direction that a

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

compass might indicate, illustrating the 360 degrees of functionality inherent in this general purpose design. And it wasn't just designed for many purposes, but also to work compatibly with any attached device or application into the future. The promise of a computer that would be compatible across models, across attached devices, and into the future created a new level set of technology which has endured ever since, with programs that were written in the 1960s often still being able to run today. As a result, customers would no longer have to rewrite or recompile applications that worked, and were able to focus on building new applications and enhancing established ones instead of constantly redoing them—a lesson that most other platforms have yet to learn.

The System/360 was designed as a family of processors from the smallest Model 20 up to the Model 195, a span in capacity from 1 to 500. All the systems could use the same external devices, disk, tapes, printers, card-readers, etc. such that an upgrade from one model to another did not force change of external devices. For an application program the same was true, no need to redesign or recompile—forward compatibility. An interesting thought in the System/360 was the idea of splitting architecture from implementation. The instruction set was the same for all models, but the implementation was different using either hardware only or hardware including micro-coded function for better performance.

Today, we know the modern version of this computer as the mainframe, and there are few other platforms that can lay legitimate claim to this designation. The term itself was coined by Gene Amdahl—the architect on the System/360—in 1964. At the time all big computers had several boxes or cupboards—frames, the most important being the box containing the processor logic—the main frame—the mainframe.

The System/360 was developed as a multi-user system offering architected separation of users using storage protection between different users, among many other definitive strengths. But many further innovations would soon move this platform forward in many ways that have never been exceeded by other platforms, from virtualization and storage to time sharing and security, and beyond.

Also at this time, IBM systems could only be leased—not bought—assuring a constant and growing cash-flow to support IBM's financial health—and assure enough funds to further develop systems of the future. This strategy proved so successful that IBM eventually had to modify it to allow for the purchase of the hardware just to ensure a fair competitive landscape.[27]

[27] Emerson W. Pugh. *Building IBM (History of Computing)* (p. 254) (The MIT Press, 2005), Kindle Edition.

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

Subsystems, Databases, TP Monitors

When System/360 was first designed, the large majority of activity on it took place in a non-interactive mode—often referred to as "batch." Batch jobs would be submitted to run on the system, generally using 80-column-wide punch cards with "Job Control Language" or JCL (which will be explained in the following chapters) wrapping the contents to tell the computer how to process them. The results would then end up in a print or storage medium or on more punch cards. But the times, they were a-changing—which was a good thing, because this coincided with the era of the "do not fold, spindle, or mutilate" movement of anti-technology protestors, whose slogan was derived from the instructions that accompanied punch cards sent to consumers to be returned to the large organizations that provided them with goods and services.

Before multiple users could concurrently use the mainframe, though, there was a requirement that multiple system tasks and subsystems also be able to run concurrently,

© Everett Collection/Shutterstock.com

Apollo Rocket Launch

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

as is appropriate for a true operating system. This need manifested at the same time as the Apollo space program approached its apex, and so there was a connection between many advances on the mainframe and the space program, particularly including the mainframe's first major database, IMS (for "Information Management System"), which was built on prior art and designed to store information in a hierarchical manner that was structured to mimic the hierarchy of assemblies and sub-assemblies and basic parts in the Apollo rockets.

Other uses that required multiple concurrent users began to emerge, and so teleprocessing monitors (aka "TP Monitors") such as CICS ("Customer Information Control System") were born, allowing many users to access the same applications and data at the same time, often using text-based video terminals called "3270" ("thirty-two seventy") based on their model number. CICS became one of the most successful software products in history, and is still in heavy use for a wide range of purposes, including running the applications used on most bank automated teller machines in the world.

One of the fullest manifestations of the idea of time sharing was achieved at the beginning of the 1970s with IBM's Time Sharing Option, or TSO, which allowed multiple concurrent users to logon to the mainframe and perform any action they wished, such as accessing system data, writing and compiling programs, and running those programs, as if they had the whole computer to themselves.

Operating Systems

Of course, in order to run a computer as more than a one-at-a-time processing environment, there is a need to have all the instrumentation and services and tasks appropriate for it to be continuously operating and taking input, processing it, and producing results. There was prior art of what an operating system should be—for example, the 1959 SHARE Operating System for the IBM 709. But, of course, for IBM's new singular line of mainframe computers, the state of the art had to be advanced, and so OS/360 was designed and written, with many lessons learned (as described by Dr. Brooks).

By the time the original great mainframe operating system, OS/360, was completed, vast amounts of time and effort and personnel had been invested in its creation. It was then discovered that, while it ran well on the largest mainframes, it was too large to fit on the smaller models, total memory, aka primary storage, being well under a megabyte on those smaller machines. So, even though all the S/360 machines could run the same program instructions, any program that was too big to fit in memory couldn't—especially if most or all of it had to remain in memory most of the time, as with an operating system.

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

So it was that DOS was born. No, not any of the personal computer diskette operating systems that ran on Intel and other consumer chip sets. This was the Disk Operating System, and it was designed to handle the basics and fit into a smaller amount of memory, at least until the customer got a larger computer and could upgrade to OS/360 or its successor. Over the years, however, users of DOS got used to it, and its lower license fees became part of the equation. So IBM had to support two different streams of operating systems. Over time and many names, OS/360 eventually became "MVS" for "Multiple Virtual Storage" and today is known as "z/OS" for "zero downtime Operating System." That "z/" prefix has shown up in the name of three other operating systems on the mainframe as well, beginning with z/VSE, which is descended from VSE, which is descended from DOS.

The next operating system started out as ACP or Airline Control Program,[28] a specialized environment for running travel-related application environments. It was eventually upgraded to TPF for Transaction Processing Facility, and today is known as z/TPF.

For many people, however, there is a special role for the last of IBM's original four mainframe operating systems: VM—the original virtual machine. Ironically, when it was created in 1972 as an elaboration on prior art such as CP (for Control Program), it was intended to enable the customers of DOS and its successors to migrate to the successors of OS/360 by allowing them to run both environments concurrently on the same mainframe while doing the migration. Nonetheless, VSE persisted.

There was a time when AIX, IBM's UNIX-flavored operating system, ran on the mainframe, but it was relatively brief.

But two other UNIX flavors subsequently arrived and stayed, the first being an addition to the successor of OS/360 known at the time as MVS/ESA, and the UNIX flavor being known as MVS/ESA Open Edition, which was essentially a set of services that allowed programs running on the mainframe to access any UNIX activity as if they were running under UNIX itself, while still running on the mainframe operating system. As a bonus, it was written from scratch to meet the POSIX standard that defined a proper UNIX, and became the first UNIX in the world to be fully POSIX compliant, despite not being a standalone operating system. Today, it is known as z/OS UNIX System Services, and handles most internet activity based on the z/OS mainframe.

The second one is Linux. While its nickname is z/Linux, its real name is Linux on IBM Z. It arrived on the IBM mainframe in 1999 as a port from its original Intel

[28] https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zmainframe/zconc_opsysztpfintro.htm

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

environment, and has taken root, often running many concurrent copies under z/VM, but now capable of running as any number of containers under Kubernetes under z/OS. Among many advantages this Linux has over Linux on other platforms, it has all the hardware strengths of the IBM mainframe, and it can do in-memory TCP/IP with other OS instances running on the same physical mainframe concurrently, giving massive speed and security advantages.

Controllers and Other Devices

The mainframe has always been a frugal environment, as the demand for its resources has always stayed ahead of the Moore's Law curve—even more so today with Moore's Law grinding to a halt. For that reason, it has always been important to preserve the main CPU for application-related activities, and allow secondary computers known as controllers to offload interaction with devices such as disk and other direct storage devices (aka DASD for "Direct Access Storage Device," pronounced "dazz' dee"), terminals, tape drives, printers, and network-type connections to other mainframes.

While the details can wait for later in this book, the essential perspective to have on this is similar to a feudal kingdom or modern bureaucracy, with the highest-level actors protected from the masses with a layer of functionaries who take care of both miscellaneous local activity and gatekeeping of access to the higher-level actors.

Inevitable Enhancements

Since its inception, the IBM mainframe has remained the most leading-edge computing platform on Earth, and today it continues to offer leading-edge advances while continuing to support and be compatible with the key functionality that was part of its initial design. What that has meant is that, as enhancements such as time sharing, virtual memory, security, terminal interaction, and even physical processor construction continued, preservation of legacy compatibility has remained an important emphasis.

Easily one of the most significant examples of this is processor address range. The original System/360 had a 3-byte memory address range (it was also one of the main reasons the 8-bit byte became the standard). That's 24 bits—it's also reminiscent of the IBM logo with eight bars in each of the letters. And it's also 16 megabytes. Well, if 640K was enough for Bill Gates two decades later, 16 megabytes certainly seemed like enough for the original mainframes. In fact, no mainframes were built that even had that much physical memory until the 1970s. But by the 1980s it was beginning to feel cramped, so an extra byte was added to the address range. Well, 7 bits to be exact,

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

because the 8th bit was used as an indicator of whether a given program was referring to memory addresses that could extend to 2 gigabytes, or was stranded under the 16 megabyte "line." The new 31-bit range was known as extended architecture, or XA, which is recognizable in the name of the then-current mainframe operating system MVS/XA.

Of course, even with virtualization, 2 gigabytes wasn't enough for long. Initially, therefore, a stopgap was put in place that would allow programs to adopt additional address spaces in memory to increase the amount of data that was treated as being in primary storage. These address spaces were called hiperspaces, in a nod to science fiction (and not the only one—remember USS?). But it was kind of clunky and not exactly transparent to applications to deal with these extra address spaces, so it became clear that nothing short of 64-bit addressability would do—which therefore happened.

Now, today, there are still programs on the mainframe that do everything "below the line"—that is, in the first 16 megabytes. Generally, these were written a long time ago—or talk to other programs that were—and may have been enhanced, but not upgraded enough to use a broader address range. Their residence mode—or "RMODE"—is set to 24, and their addressability mode—or "AMODE"—is also 24.

And there are programs with 31-bit residence and/or addressability—though sometimes they have to remain resident under the 16-megabyte line just so other programs of that ilk can talk to them.

But if the division between the first 16 megabytes and the rest of memory is called the line, you may well ask what the name of that invisible 2 gigabytes between 31-bit addressability and 64-bit addressability is. To which the answer is: "the bar." And more and more subsystems and applications are taking advantage of this vast amount of memory (16 exabytes), while retaining compatibility with earlier programs in whatever manner serves best.

The People

Early Mainframers

In the beginning were the users, and they used non-electronic systems to achieve their business, academic, military, and other goals. And as electronic computing became available, and user organizations such as SHARE consolidated input and solutions for how manufacturers could adapt their products to better serve needs, various roles began to emerge. But the first role, the one that truly drove the advancement of electronic computing, was always the user. And that user was often an

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

accountant—so much so, that early Data Processing (or Information Technology or Information Systems) departments were often part of the Finance department in many organizations.

These users were sometimes "power users" who could wield a certain amount of technical prowess over the electronic environments that served them, but over time, those who had technical abilities tended to diverge into specifically technical roles.

Meanwhile, there were also the people whose roles were specifically to design and create the systems—from architects to analysts to programmers, along with various hardware experts. And these people often worked on maintaining the systems as well, so they tended to group together.

Then there were those whose job was the day-to-day operations of electronic computers—operating paper or screen-based consoles, mounting and dismounting tapes on tape drives, and even making changes to DASD configurations—not to mention acting as a help desk.

And of course there has always been the management and supervisory class that interacted with all of these areas.

And there have been the IBMers (and eventually OEM and ISV representatives as well) who acted as interfaces between the originator of the technology and the user organizations.

Over time, the design-and-create people diverged into a wide range of specialties, while the operations people became responsible for a growing list of day-to-day functionality, and the users continued to be the true customers of all this activity. While each mainframe environment is unique, certain commonalities have emerged about most people in these roles over the decades, and while there has been waxing and waning of various responsibility sets, there has also been the emergence of somewhat standardized jobs.

Specialties Emerge

While no comprehensive list can be made, and it is unlikely that any environment has all and only the roles described below, this gives a sense of a selection of some of the areas of responsibilities that have emerged over the decades in a typical mainframe shop.

Application User

Can create and manipulate data using application programs and utility programs, but does not create programs.

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

Application Programmer

Part of the application development area, with job titles that may range from Junior Programmer to Senior Programmer Analyst and Systems Analyst, all of them are responsible for the creation and maintenance of the applications, which are often written in COBOL or specialty languages.

Operator

Operators are responsible for the day-to-day monitoring of the mainframe environment and responding to emergent requirements that are documented. Systems automation of the operating system and physical automation of things such as tape in silos have reduced certain types of activity. However, operators still remain engaged in help desk activity and monitoring for emergent problems that automation can't solve.

Performance and/or Capacity Analyst

On the mainframe, understanding the performance characteristics of the environment is essential to planning for capacity upgrades. You never just add another box, and rarely just add more memory or CPU capacity without a planned approach. Part of this is often using the System Management Facility (SMF) data that is generated by much of the activity on the mainframe. These data give valuable insights into the amount and nature of the mainframe's activity, and are also often used for chargeback, to ensure the heaviest users of the mainframe pay their fair share of the costs.

Security Administrator

Sometimes, this person's main job is to give security access as requested, including creating and deleting user ids on the system. Sometimes this person is more technical and/or in an architect role, and is responsible for planning and implementing security enhancements. And sometimes, they're a user who just takes care of the security for their own user department in coordination with centralized computing security personnel.

In any case, the weight of keeping the mainframe the most secure environment available is on their shoulders, and it is a burden they need to deliberately share with the rest of the organization, whether through education, or coordinated planning of maintenance, configuration, and acquisition of additional security features.

System administrators/Programmers

These are the most technical people, responsible for the installation and maintenance of the operating system and any other software products that operate at that technical level. They are also the last line of troubleshooting when things break. And they tend

ALL RIGHTS RESERVED

Chapter 1 Evolution of the Modern Mainframe—How Did We Get to Where We Are?
An Exerpt from Introduction to Enterprise Systems by Harbeck, Boyes, Stenfors, & Seay
9781792483042 | 1st Edition | 2022 Copyright | Property of Kendall Hunt Publishing

to have a deeply technical approach to life that often accompanies such a deeply technical job description. They are often treated with a stand-offish awe.

Because of the power they wield, it is very important to have strong technical security or auditing personnel who can act as a counterbalance to them, ensuring that they are kept above suspicion.

Network Specialists

Often included with the Systems staff, these folks also work closely with distributed network personnel, and in smaller shops may be the same people.

Database Administrators (DBAs)

One of the main ways that mainframes process data is by employing massive databases, and while IBM's Db2 is perhaps the most common such database on the mainframe, there are many others, including the older IMS (see above), Broadcom's Datacom and IDMS, Data Kinetics' tableBASE, Software AG's ADABAS, and even Oracle and other distributed databases that run on the Linux environment.

Production Control

Whether planning and configuring regular backups, ensuring that nightly application batch jobs run, or any number of other day-to-day activities, these are the folks who keep the mainframe running from the perspective of user applications and data.

Conclusion

We have offered an explanation of how the modern business computing platform evolved from a cultural and historical context. Our next Chapter will go into considerable detail as to the design philosophy of this technology.

KH ALL RIGHTS RESERVED

KH
ALL RIGHTS
RESERVED