## Chapter 4: A Brief History of Text and the Computer

### *1. The central role of text in the development of the computer*

The first programmable digital computer was built in the 1940s. The processing of text is an even more recent practice, dating from the 1960s. It was only in the 1990s that the computer also began to offer a serious alternative for the *distribution* of texts. Yet in less than half a century computers have insinuated themselves into the texture of society to such an extent that it could not function without them. This chapter examines how the computer has in such a short time come to play such a predominating role specifically in the textual world.

Some major milestones in the development of what has since become the digital textual medium can be identified.
In the nineteenth century the single-purpose calculating machine, was first conceived to have the potential to be turned into a Universal Machine, capable of performing tasks that may be expressed by way of an algorithm. In the 1940s the first Universal Machines were built. In the 1960s the computer as a Universal Machine was enabled to process text, which gave it a role in the text creation phase. In the 1980s the graphical man–machine interface of the computer greatly enhanced the possibilities for the typographical rendering of text. It enabled the computer to play a central role in the production of printed matter. The graphic interface also paved the way for the co-existence of two different ways to treat text digitally: the logical and the typographical.

- In the nineteenth century the single-purpose calculating machine, was first conceived to have the potential to be turned into a Universal Machine, capable of performing tasks that may be expressed by way of an algorithm.
- In the 1940s the first Universal Machines were built.
- In the 1960s the computer as a Universal Machine was enabled to process text, which gave it a role in the text creation phase.
- In the 1980s the graphical man–machine interface of the computer greatly enhanced the possibilities for the typographical rendering of text. It enabled the computer to play a central role in the production of printed matter. The graphic interface also paved the way for the co-existence of two different ways to treat text digitally: the logical and the typographical.
- In the 1990s the computer was included in a network, which enlarged its role as a communication tool from that of an aid for the production of analogue printed matter to a new, fully digital medium in its own right, also comprising distribution and consumption.

In his ‘communications circuit’[1] Robert Darnton has conceptualised the entire transmission process of books and other printed text forms as it has functioned for several centuries. This model visualises a process in which various consecutive actors work

[1] See Robert Darnton, ‘What is the History of Books’, *Daedalus*, Summer 1982, pp. 65-83.

together under varying cultural, economic, and political conditions to disseminate an author's text so that it can reach its readers. Laying the communications circuit over the transmission process of digital text this model will identify the similarities and differences (or continuities and discontinuities) between the new digital medium and its predecessor the print medium.

The process of making texts public and disseminating them comprises various distinct stages. These roughly correspond to those identified in the communications circuit: the creation of the text (writing), followed by its production (multiplication), distribution (the moment the text is made public), and finally consumption (reading). However, to inspect the different stages in the development of the role of the computer in the transfer of text more closely, I would like to propose one small adaptation to this chain of stages. I would like to place a magnifying glass over the first link in the chain, the 'creation stage', which is the the phase in which the contents and form of the text have not been finalised. Besides the writing of the text by the author this phase also comprises its editing, whether this is done by the author or by someone acting on his or her behalf (for example the publisher's editor). Technically, this means making a distinction between (a) text entry, (b) text recording, and (c) the manipulation of the text once it has been entered. Recognising this fluidity in the creation stage, comprising writing and editing in any number of iterations, makes it easier to trace the development of the computer's role in the writing process. Roughly three stages in that development can be recognised. These partly overlapped, but they are fundamentally different enough to treat them separately. The stages are (1) the representation of text on the computer (entry, recording, storage), (2) the manipulation of stored text for scientific and professional applications, and (3) the actual word processing on the pc, as an aid in the authorial thinking and writing process.

Among the most popular computer applications today are no doubt chatting, word processing, emailing and Web browsing, all text-based pursuits. But also outside of these text applications text is the key to our computer use. In all arithmetic, analytical, medial and other applications for which the computer as a universal machine lends itself, text has a central place. On the World Wide Web—and on the Internet in general—text is the most common way to organise, search, and find information, even when that information itself is not a text but, for example, a music file or an image. In all daily dealings with the computer text furnishes the chief interface, of the operating system as well as the applications. Files are named and stuck in folders, which are again named using text. But also beyond this kind of daily consumer use language is the basis for all human–machine interaction. All modern programming languages use a form of natural language. Also markup (one of the most important ways to encode text on the computer—and the technical basis of publication on the World Wide Web) is an entirely textual practice.

In the previous chapter I described how western society is shot through with the social and cultural significance of books as the main means to transmit knowledge. I have called this the Order of the Book. Against this background it seems only natural, and in fact almost inevitable, that the computer was to be deployed for textual communication as soon as this became possible, and that the whole human–computer interaction became a textual affair. Indeed, the eagerness with which the word processor was embraced in the 1980s seems to confirm that idea. Given the prominence of text-based applications in popular computer use today, the question even presents itself why the computer was invented as a calculating machine rather than a language machine. As it is, the computer continues to have to recalculate all those textual data and instructions that we feed it to the only meaningful units which it knows: ones and zeros. Why would it not be possible to calculate with language itself? The idea may seem stranger than it is. His whole life Wilhelm Leibniz continued to believe in the construction of a language consisting of logical symbols that could be manipulated by means of a calculator. Such a language, and a machine to 'calculate' it would enable any philosophical debate to be settled with the click of a button.[2] That Leibniz' dream has still not been achieved, is not so much because such a logical system of symbols is not viable.[3] The real problem is that the subtle shades of meaning we can—and want to be able to—express with natural human language are simply not amenable to being reduced to a system of logical symbols.

Zeros and ones it was, then. For the sake of convenience, however, it was felt necessary to devise a way to cast instructions to the computer into a humanly intelligible shape. Hence program lines, menus, file names and the like now all have a human-readable form, even if behind the scenes the computer still calculates with the only numbers it knows: ones and zeros. No user now stops to think that every keystroke is converted into a series of binary numbers. In fact, in our perception language is the primary way in which we deal with the computer today. The numbers that the computer really crunches appear to play no more than a subordinate role; the numbers seem to dance to the tune of the text. But once the reverse used to be the case and, thinking from the binary heart of the computer, the quest was for a way to represent letters.

Given the enormous importance of text for average daily computer usage it is striking how much effort it still took before the computer could actually deal with text. How did that process take place and why did it take so long? What factors impeded and stimulated it: design and chance, unintended effects, failure of intended effects, etc. This chapter will reconstruct that process in general outline.

That text has come to take a central position on the computer appears at first sight to be only natural—a reflection of the importance of text in society. At the same time some commentators point out that text is actually beginning to lose its prominence.[4] They are obviously not suggesting that we are about to engage in a direct binary data exchange with the computer, or that humans have recently acquired massive training and experience in symbolic logic. What they mean is that in addition to text, other modalities, especially images, are playing an increasingly important role in digital communication, as in society at large. This is often referred to as the ascendancy of visual culture.[5] One simple explanation for that increase of other modalities could be that the digital medium makes it easy, as a result of the convergence identified in Chapter 2, to integrate modalities such as

---

[2] In *The Courtier and the Heretic: Leibniz, Spinoza, and the Fate of God in the Modern World* (New York and London, 2006, p. 79), Matthew Stewart gives an account of this ideal of Leibniz.

[3] Alfred North Whitehead and Bertrand Russell's *Principia Mathematica* (1910-1913) is impressive evidence that it is, even if Douglas Hofstadter is right with his interpretation of the implications of Kurt Gödel's explosive article in 1931 for the fate of Russel's fortress, which he deemed impregnable (see Douglas Hofstadter, *I Am a Strange Loop*, New York, 2007, Chapter 10).

[4] Steve Johnson, *Interface Culture: How New Technology Transforms the Way We Create and Communicate* (New York, 1997, see pp. 148-52), is one of the exceptions.

[5] See, for example, Mitchell Stephens, *The Rise of the Image, the Fall of the Word*.

images and sound in text. But the notion of a visual culture is not that new, and certainly predates the advent of the computer. From the beginning of the twentieth century in many places in the world all kinds of visual languages have been designed for signs, packaging and other forms of communication.[6] In the middle of the last century De la Fontaine Verwey finds in his contribution to *Copy and Print in the Netherlands* that the image, '[s]uperseded for a time by the book', 'has resumed its ancient rights and is engaged in fulfilling tasks that have for centuries been carried out by the printed word.'[7]

Not only are the signs that text is beginnning to lose its prominence still rather faint, the role of text has probably simultaneously been strengthened in other ways, such as the largely textual interface of the computer and the Internet, but also the extraordinarily popularity of texting on the mobile phone. To judge by the popularity of social networks, blogs and the comment function on so many websites, it may well be the case that more people write—at least with a form of publication in mind—than ever before. The phenomenon is not necessarily always equally visible, however. An example of a less directly visible use of text is the way keywords are assigned to images and sound in order to be able to search for them. This may be a transient phenomenon while the searchability of images and sound through other images and sound is still in its infancy. For the time being at any rate the entire digital world—including games and chatting—is accessed by means of text.

If the relationship between text and other modalities is indeed changing, the change, at least so far, seems not particularly drastic. Nonetheless, in a longer historical perspective a situation may well be imagined where text need not necessarily be the most important means of communication. I will return to this speculation in Chapter 6.

## 2. The history of computers and computing

There was initially little evidence of the important role that text was to play in the digital world. The history of the advent of text to the computer starts with two major developmental leaps in the history of the computer itself. Two in particular are important: (a) that from machines with only one function to multifunction machines, and (b) that from mechanical to electronic, digital machines. In the category of machinery with only one function, two are of particular relevance to the history of the computer as a machine for the processing of text. The first is the calculator, which still forms the heart of every computer. The second is the typewriter, which delivered, in the shape of the keyboard, the chief means of input for the computer today. In addition, there is a number of more specialised machines, some of which I will also briefly mention.

The history of the calculator as a forerunner of the computer goes back some four

[6] There was an enormous belief in the potential of images (in the form of pictogrammes and icons, but also image-based statistics) in promoting efficient information transmission. A particularly prominent and tireless advocate of the use of information graphics was Otto Neurath, the inventor of the Isotype (International System of Typographic Picture Education) symbols in the 1920s. After fleeing his native Austria in the 1930s he founded the International Foundation for Visual Education in The Hague, and later the Isotype Institute in Oxford.
[7] H. de la Fontaine Verwey, 'The twentieth century', in W. Gs Hellinga, *Copy and Print in the Netherlands: an Atlas of Historical Bibliography*, Amsterdam, 1962, pp. 59-67, on p. 59.

centuries. In 1623 Wilhelm Schickard (1592-1635) from Tübingen made a 6-bit 'counting clock', which could add and subtract. He called his machine a clock because the machinery was reminiscent of one. The instrument was entirely mechanical. When half a century later Wilhelm Leibniz began to work out his idea for a digital calculator, he was a great deal more ambitious. His machine was to be capable of processing universal logical symbols. In spite of his unbridled ambition and dedication, he never managed to go beyond a kind of mechanical pocket calculator which could add, subtract, multiply, and divide. Like Leibniz in the seventeenth century, the British mathematician Charles Babbage in the nineteenth century had the vision that calculators could be used for purposes other than making numerical calculations. In the intervening centuries scientific knowledge and instrument making skills had advanced so much that Babbage was able to take the implementation of his ideas further than his predecessors. Although Babbage never built more than parts of his 'Analytical Engine',[8] on the strength of his design he can be considered as the creator of the first Universal Machine. Like the calculators of Schickard and Leibniz, it was entirely mechanical (it was to be powered by steam) and made use of decimal instead of digital numbers, but it was programmable, separated the data from the program, and was capable of loops and conditional branching. That was more than most computers were capable of even a century later. Babbage was even considering exporting the outcome of calculations to punched cards. This notion, inspired by the Jacquard loom, would have enabled the machine to write and store its own programs.

Charles Babbage had the vision; Ada, Countess of Lovelace, a fellow mathematician who heard him expound on it one night over dinner, was one of very few people who understood its implications. Recognising that on a higher level of abstraction computing was not counting but the manipulation of symbols, she proceeded to devise a number of algorithms that might actually have been executed by the Analytical Engine had it ever been built. When Lovelace translated an article on the Analytical Engine by the Italian mathematician and military engineer Luigi Menabrea she added some very percipient notes of her own, amounting to twice the length of the original article. In these notes she correctly predicted that a machine like the Analytical Engine might be used to compose music, produce graphics, and perform a variety of scientific tasks:

> [I]t might act upon other things besides *number*, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine. Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.[9]

---

[8] Originally Babbage had designed a simpler 'machine', which he named the 'Difference Engine' because it was able automatically to generate tables of the intervals (or differences) between sets of numbers resulting from programmed series of progressive additions. The machine could produce a print of the tables.

[9] 'Sketch of The Analytical Engine Invented by Charles Babbage, Esq.', by L. F. Menabrea, with notes by Ada Lovelace, *r*eprinted in Charles Babbage, *Science and Reform: Selected Works of Charles Babbage,* ed. Anthony Hyman, CUP, 1989, pp. 243-311, on p. 270. Emphasis in the original.

Evidence that the vision of Babbage and Lovelace could become reality, was only delivered by Alan Turing in the middle of the twentieth century. According to Turing his abstract 'Turing machine' was capable of executing all functions which can be calculated in the form of an algorithmic procedure. Modern digital, electronic programmable computers which met Turing's requirements were first developed in the 1940s.

The binary principle was used not only for the calculations themselves, but also for the way in which the data were encoded. Just as numbers can be represented both in a binary system and in a decimal one, the same is true in principle also for text, image and sound. In the case of numbers and text the number of discrete characters is very limited, and each character can be represented by a limited number of bits. For Latin script, a single byte (eight bits) can encode 256 unique characters. Modalities like image and sound are more complicated to encode. Here the signal has to be divided into any arbitrary number of constituent components. Dividing an image (or sound) into discrete particles means that transitions will never be continuous, but always incremental. The number of components per unit of signal (for example, pixels per inch) decides the realism of the binary representation: the more the better. But however high the number of pixels per inch, the realism of a digital rendition can in principle never be equal to an analog rendition. In spite of all its shortcomings the relevance of binary representation is that all data in all modalities and all the calculations that could be applied to them, can be encoded in the same binary fashion. This makes 'binariness' the 'element'[10] in which the much-vaunted convergence of modalities (on which more in Chapter 5) can take place.

The typewriter is the second single-function machine besides the calculator that has been of great importance in developing text encoding on the computer. Some of the earliest typewriters were designed for the blind,[11] which nicely illustrates how deep the divide can be between an inventor's intent and the actual social use of an invention. Of special interest for the present topic is the case of the keyboard. Of all the ingenious typing systems ever designed[12] it was that by Christopher Sholes, the creator of the first typewriter to be taken into commercial production, that became the standard. This was the keyboard with the well-known qwerty layout.[13] The most important legacy of the Sholes keyboard is that the characters found on his keyboard are now still the atomic building blocks of text on the computer. The standard computer keyboard has no accented characters, makes no distinction between a hyphen and an em-dash, or between the decimal point and the full stop, and lacks all sorts of special characters: from typographical through mathematical to currency signs. Instead it was visual appearance only that decided whether a separate character was created.[14] The computer keyboard encodes individual letters binarily and enters them into the computer. Just as on a typewriter, this is done by assigning a single character per key, although that number may be increased by using the shift key (and on the computer in addition various function keys).

---

[10] The term is Michael Heim's, from *Electric Language*, p. 102.
[11] Michael H. Adler, *The Writing Machine*, London, 1973, p. 48.
[12] Adler, *The Writing Machine*, p. 25-90.
[13] The qwerty layout is still in use in many countries, for example, throughout the English speaking world. In some other countries the layout differs. France uses the azerty keyboard, while Germany and some Eastern European countries use the qwertz keyboard.
[14] Hence on some keyboards no separate figure 1 was included, the letter l being regarded as suffiently similar in shape.

Among the many inventive and less inventive alternative text entry systems that did not make it must definitely be mentioned the idea of Douglas Engelbart, also known as the inventor of the computer mouse,[15] to enter the 31 characters of the standard 5-bit code of the 'teletype' (the forerunner of the telex) by the simultaneous pressing of five keys ($2^5 = 32$).[16] Engelbart worked on this 'five-key handset' in the 1960s, as part of his ambitious Framework for the Augmentation of Man's Intellect framework, which will receive more attention later in this chapter. While the idea was not new (in the course of the nineteenth century several typewriters with piano-type keyboards had already been designed) and certainly had advantages, it was not up against the domination the qwerty keyboard had by then already acquired. It was in use in large parts of the world, and generations of typists had learned typing blind using the qwerty layout.

The typewriter did not take the process of creation – editing – production – publication – distribution – consumption beyond the creation stage. It took care of the 'data entry' and 'storage' (anachronistic terms for functions that were really only created by the computer) of a text, but could do little for its reproduction, publication and distribution.[17] As a medium this does not distinguish the typewriter substantially from manuscript—with the exception perhaps of the degree of readability. In that regard the typewriter only very partially approaches printing type. That did not, however, keep its inventors from stressing this property, even in the case of the very earliest machines.[18] The magnifying glass that I placed over the creation and editorial phase shows that the typewriter is a rather poor performer when it comes to the manipulation of text.

Among the more specialised techniques that are relevant in the history of the computer certainly belongs telegraphy, and in particular the Baudot system for text input dating from 1874. That is the standard 5-bit code already mentioned in the discussion of the keyboard. Despite the limited number of characters (a maximum of 32) that could be encrypted with this five-bit system, this character encoding by Emile Baudot (1845-1903) still remained in use in the digital electronic environment until it was replaced by ASCII in the middle of the 1960s.[19] Since the morse system with its dots and dashes is also binary, Baudot's encoding system lent itself particularly well for transferring to the computer.

Another specialist device for the processing of text that deserves attention was the typesetting machine used in in print production.[20] The typesetting machine used the typewriter keyboard to its advantage (albeit that the typesetting machine's keyboard was equipped with substantially more keys; the Monotype had four complete sets of qwerty keys, one each for roman, italics, bold, and small caps). At least four major improvements

---

[15] See Thierry Bardini, *Bootstrapping: Douglas Engelbart, Coevolution, and the Origins of Personal Computing*, Stanford, 2000, pp. 81-102.
[16] See Bardini, *Bootstrapping*, pp. 58-80.
[17] Except of course with the rather limited help of the carbon copy. The stencil machine and the duplicator can be disregarded. While these techniques make use of a typewriter to record the text, multiplication is a separate step, which requires a duplicating machine.
[18] 'A British engineer, Henry Mill, was granted British patent no. 395 in 1714 for a device capable of impressing letters on paper or parchment one after another "as in writing", the product being so neat and exact as to be indistinguishable from printing' (Adler, *The Writing Machine*, p. 47).
[19] Bardini, *Bootstrapping*, pp. 65-79. More will be said about ASCII (American Standard for Information Interchange) later in this chapter.
[20] Other, more marginal systems that can be mentioned were for example those used for the generation of titles and subtitles for film and television, which were also becoming more sophisticated as time went on.

which were applied early if not for the first time in the typesetting industry, have been of great importance for the development of digital word processing. These were the use of a storage medium, in the form of the punched tape of the Monotype typesetting machine from 1887; the application of the tele typewriter for remote typesetting through the 6-bit code of the TeleTypeSetter (TTS) in the late 1920s; the application of the computer in the third generation of phototypesetting machines from the late 1960s, and the development of the concept of markup in the 1960s and 1970s, about which more later.

## 3. History of word processing

We have become thoroughly used to the fact that in the digital electronic world all modalities can be rendered in a single medial environment. To make this possible, the calculator, as a forerunner of the computer, had to evolve from a machine to answer arithmetical questions to a machine that could perform calculations in which the aim was not the numerical outcome, but the manipulation of numbers as symbols. In word processing these symbols stand for the characters that make up a text, but also for a variety of codes that define the typographic representation of those characters. Simply put, the history of word processing is the history of how the computer as calculator married the typewriter, stimulated by the inspiring examples of mechanical 'word processing' offered by the telex and the typesetting machine. But even though it is tempting to write history into the present, this is a too simplistic representation of the facts. Despite the advanced ideas of Babbage and Lovelace in the nineteenth century, the *rapprochement* between the computer and the typewriter in fact took quite some time. If for example Turing and other computer pioneers had such non-numerical use of computers in mind, it was certainly not for the processing of text in the present sense of the term.

Reasoning back from the present, word processing has the three ingredients that became visible under the magnifying glass at the beginning of this chapter. In order of their development these are: (a) text entry, (b) storage for reuse and (c) manipulation of the entered text. The first of these is the least interesting: that is what the typewriter had already been doing very well for over a century. The distinction between 'text entry' and 'storage' is obviously something of an anachronism: the typewriter did both simultaneously. But 'storage for reuse' will transpire to be fundamentally different. (The three above-mentioned functions play at the level of individual texts. Above that level there is also the function of information management: organising access to texts. That aspect, too, is obviously of great importance in the context of the history of the computer as a textual medium. I will return to it under the heading 'Document management' below.)

Even when the computer could handle text it took a long time before the implications of this fundamental distinction between text entry and storage for reuse—and the opportunity this offered for further manipulation—became really clear. The word processor as it was developed in the course of the 1980s is based on that slow recognition of the implications of the ability to alter and reuse text even during the creation process. A major breakthrough was therefore the integration between the two distinct levels of text entry and editing. That this distinction was initially often made between them betrays the origin of word processing in the office environment. It shows that insight into the real possibilities of word processing dawned only slowly.

*Mechanical* The typewriter as it was invented and improved in the course of the nineteenth century could only 'enter' and record text. Once the text was recorded ('inscribed' on paper) reuse was not possible except by retyping it, and correcting was only possible by crossing out previously typed text and adding new text with pencil or pen. The earliest example of reuse (and to a lesser extent, correction) of text after it was entered is that of the already mentioned punched tape of the Monotype typesetting machine from 1887. The output of Monotype keyboards was solely to the punched tape, which in turn served as input for a casting machine. The purpose of employing punched tape was not primarily reuse but separating two very disparate activities in the labour process, enabling more efficient plant use. The automated casting took less time than text entry: about one and a half to two typesetters were needed to keep one casting machine employed. Nevertheless the result of thisi functional separation was that composed type no longer needed to be preserved for a possible reprint but could be set again from the punched tape. Compared to storing set type this also saved a great deal of physical storage space. This system of separating text entry and storage through a punched tape has been much imitated. In telegraphy, for example, it was used to be able to send the same message to different recipients.

*The computer* Even before the Second World War the first electric typewriters could be found in offices, but in order for re-use and manipulation to be made possible the computer was needed. It was the Second World War that gave a major impetus to its development, for example by the pressing need for accurate ballistic calculations. Experiments with new technologies and new architectures took place at a feverish pace, and the period between 1941 and 1948 saw tremendous progress. From this short period date all the famous forerunners of the modern computer: the Zuse Z3 by Konrad Zuse in Germany; the Atanasoff–Berry Computer by John Vincent Atanasoff and Clifford Berry in the United States, the Colossus of Tommy Flowers and others in the United Kingdom; the Automatic Sequence Controlled Calculator by Howard H. Aiken of IBM; and the Electronic Numerical Integrator And Computer (ENIAC) by John Mauchly and J. Presper Eckert, the latter two both from the United States. The British 'Bombe', too, co-designed by Alan Turing to decipher encoded enemy messages, was an example of the enormous computing power that was being generated for the war effort. In the present context the Bombe, despite its limited functionality, is especially interesting because the symbols manipulated by this electromechanical device represented letters, perhaps for the first time in computer history.

As a result of all these military and government exertions the capabilities of the computer slowly began to be recognised more widely. In 1949 the Italian philosopher and Jesuit Father Roberto Busa conceived the idea of using a computer in a scholarly project that involved text. Busa completed this massive pioneering project, a concordance of the work of Thomas Aquinas, after working on it for 24 years. (It is not unthinkable that Arthur C. Clarke's famous story 'The Nine Billion Names of God' of 1953 was inspired by Busa's project.) His Index Thomisticus has gone down in history as '[t] he first electronic text project in the humanities'.[21] Even if Busa's use of the computer in the avant-garde of digital word processing was primarily computational in nature, like the deciphering of German secret messages it concerned the manipulation of symbols (representing text) as conceived by Babbage and Lovelace a century earlier.

However challenging as computational problems, the cracking of codes and creating of concordances were ultimately quite straightforward functions compared with the present-day capabilities of computers. In the early 1960s computer pioneer Douglas Engelbart had much grander expectations of the relationship between text and the computer. In his essay 'A Conceptual Framework for the Augmentation of Man's Intellect' from 1963 he formulates his concept of an advanced form of text manipulation as an aid to thinking.[22] Engelbart was imagining a 'writing machine' whose description begins to sound faintly familiar to routine users of the modern word processor:

> [L]et the reader consider an artifact innovation appearing directly within the relatively low-order capability for composing and modifying written text, and see how this can affect his hierarchy of capabilities. Suppose you had a new writing machine—a high-speed electric typewriter with some very special features. You can operate its keyboard to cause it to write text much as with a conventional typewriter. But the printing mechanism is more complicated; besides printing a visible character at every stroke, it adds special encoding features by means of invisible selective components in the ink and special shaping of the character.
>
> As an auxiliary device, there is a gadget that is held like a pencil and, instead of a point, has a special sensing mechanism which can be moved along a line of the special printing from your writing machine (or one like it). The signals which this reading stylus sends through the flexible connecting wire to the writing machine are used to determine which characters are being sensed, thus causing the automatic typing of a duplicate string of characters. An information-storage mechanism in the writing machine permits you to sweep the reading stylus over the characters much faster than the writer can type; the writer will catch up with you when you stop to think about what word or string of words should be duplicated next, or while you reposition the straightedge guide along which you run the stylus.
>
> This hypothetical writing machine thus permits you to use a new process of composing text. For instance, trial drafts can rapidly be composed from rearranged excerpts of old drafts, together with new words or passages which you insert by hand typing. (pp. 6-7)

Here we find the functionality of text manipulation during the composition process clearly articulated for the first time. That Engelbart's flight of fancy resembles the idea of a word processor is hardly startling from a twenty-first century perspective. What is surprising is

---

[21] Susan Hockey, *Electronic Texts in the Humanities: Principles and Practice*, OUP, 2000, p. 5. For a definition of humanities computing see Willard McCarty, *Humanities Computing*, Basingstoke and New York, 2005.

[22] Douglas C. Engelbart, 'A Conceptual Framework for the Augmentation of Man's Intellect', in *Vistas in Information Handling*, vol. 1, *The Augmentation of Man's Intellect by Machine*, eds. Paul W. Howerton and David C. Weeks, Washington and London, 1963, pp. 1-29.

that the concept was apparently still so new in 1963. Engelbart was looking for a way to use language, in combination with certain technological 'artefacts', methodology, and training, for 'increasing the capability of a man to approach a complex situation, gain comprehension to suit his particular needs, and derive solutions to problems.'[23]

Like the rest of the volume in which it appeared Engelbart's article is an example of remarkably advanced thinking. From a present-day vantage point, however, Engelbart is providing a needlessly complicated technological solution to something that is regarded as a basic and rather obvious functionality. It is tempting to wonder why he made it so difficult for himself with his special ink and optically readable characters.[24] Apparently Engelbart was not aware of the tremendous progress that was already being made in the office environment. Obviously Engelbart's way of thinking resembled that of Leibniz more than that of an office clerk: like Leibnitz, he was a scientist. But what most people do with text on the computer today resembles the work of a secretary more than it does the scientific ambitions of people like Leibniz or Engelbart. In the world of office machines there was obviously less consideration for scholarly needs and such ambitious ideas about the computer as a tool for thinking as Engelbart's were certainly not being addressed there. But the office environment did produce very concrete results. It was office automation that ultimately yielded a decisive contribution to solving Engelbart's technical problems, even though it meant a radical simplification compared with the functionality that Engelbart had in mind.

The first machine to be actually called a 'word processor' was a very mundane machine intended for office use, the IBM Magnetic Tape Selectric Typewriter of 1964. It was followed in 1969 by the Magnetic Card Selectric Typewriter (which could also be used as a computer terminal). Such single-function machines that could only perform word-processing remained on the market at least until the beginning of the 1990s. The early word processors still did not make text manipulation very easy. The office environment was mainly focused on the basic input and output functionality, and much less on the manipulation function. In many respects these word processors resembled very much the way typewriters worked. For example, separation of the text entry and editing functions long remained a feature of word processors. This is quite likely attributable to the passive role of the office typist, who was not required to think, but simply to record a text faithfully from manuscript or dictation.

The office environment may not have had much of an eye for Engelbart's ideals, but conversely Engelbart, who was so committed to using the computer in the process of thinking—as a knowledge instrument—quite overlooked the everyday opportunities that word processing was to offer. Regarding the computer as a manipulator of higher-level symbols to be able to recognise more mundane uses, it was simply too big a step from the creation of a machine that could calculate mathematical questions and be deployed in heavy thinking tasks to a machine that could perform such simple operations as importing, storing, and moving pieces of text.

After the invention of the word processor, it still took some time before the dedicated appliance was fully overtaken by word processing on the (personal) computer.

---

[23] Engelbart, 'A Conceptual Framework', p. 1.

[24] Such optically readable characters characters were in actual use, for example in banking, for a long time.

Without doubt the most important development in the processing of text was that the emphasis gradually shifted from the capture of a 'correct' text, to the creation, during the thinking process, of a new text that did not yet exist. Though the result still fell short of what Engelbart had envisaged in the way of a 'knowledge instrument', it was no doubt an improvement. That improvement was less technological in nature than conceptual. Determining a 'correct' text was, implicitly or explicitly, based on a real or assumed ideal-typical entity (such as a manuscript, or a verbatim record of speech, or the *oeuvre* of Thomas Aquinas). Of course, the possibility of correction which a word processor offered was desirable to the extent that it made it easier for users to achieve the ideal of an error-free text. But further manipulation of the text such as Engelbart had in mind was not required in this kind use. What was needed was a recognition of the principles underlying the possibilities of 'storage for reuse' and 'correction' as they were already in use. The reuse of existing text fragments, moving and editing the text and inserting new fragments during the composition process were actually existing possibilities; it was just that the spirit of Engelbart did not yet inhabit their execution.

Examining the properties of the present-day word processor, it can be concluded that a latent need for it must have been present at an early stage. This was certainly not widely recognised. Among the few who had a vision of its possibilities was, beside Engelbart, also a young sociology student by the name of Theodor Holme Nelson. With the programming project that Nelson started as an MA student at Harvard University in 1960, he had in mind nothing less than the creation of a word processor with exactly the kind of functionality that the world is now accustomed to. What he wanted was 'a text-handling system which would allow writers to revise, compare, and undo their work easily.'[25] Unfortunately, though Nelson had the vision, he lacked the means to realise his idea. A few years later, like Babbage, he abandoned his first project to address an even more ambitious one: the concept of hypertext.

Hypertext, the term Nelson coined around 1965, certainly spoke to the imagination. The concept brought together the thought process and the writing process, combining the two in a single system capable of identifying connections between thoughts expressed in fragments of text and all kinds of other materials. In this way hypertext, like the ideas of Engelbart, went far beyond mere word processing. The purpose was for computers to meet the needs of humans to organise and link in a meaningful way both the sources of their knowledge and its reflection in their own texts. In this description the hyperlink seems no more than the digital development of the footnote. This fails to do justice to its concept if not its execution. (To Nelsons grief, the public implementation of hypertext in the World Wide Web stayed far behind the functionality he had planned.) The dynamic possibilities of hypertext as envisaged by Nelson are much greater than those of the footnote. The close way, for example, in which composition and presentation could be matched was designed to make hypertext a potentially very powerful tool in (the representation of) the thinking process. It has been suggested, for example, that using hypertext Ludwig Wittgenstein would have been better able to shape his thinking and composition process.[26] In any case,

---

[25] The programming project took place in the context of a computer course for the Humanities. See Christopher Keep et al., 'Ted Nelson and Xanadu', http://www2.iath.virginia.edu/elab/hfl0155.html.

[26] Jos de Mul, 'Wittgenstein 2.0', in *Philosophy of the Information Society*: Vol. 1, *Proceedings of the 30th*

hypertext would have been able to do better justice to the not very linear way of reading required by Wittgenstein's philosophical writings. Clever though Nelson's idea of hypertext may have been, as in the case of the word processor, it proved a more sophisticated application than what the ordinary computer user was looking for.

But there were also other incentives apart from the ones coming from the market for office machines and, to a lesser extent, the need for thinking machines in science. There was for example the quest for a good way to improve the two-dimensional output of the typesetting machine, for the benefit of offset lithographic printing. The computer proved an excellent tool for programming word break routines and to calculate visually pleasing word spacing, and in the early sixties it was introduced in typesetting machines.[27] By the 1970s the so-called 'third-generation' typesetters were fully computerised, and used digitally stored characters, which could be electronically manipulated in a variety of ways.

Another major impetus came from the computer world itself. This was the need to enter instructions and data in computers not at the binary machine level (which was extremely laborious and error prone), but at a higher level of abstraction. To this end hexadecimal keyboards were deployed, which could handle clusters of binary numbers simultaneously. Also the need to communicate with other users away from the mainframe resulted in a growing importance being attached to text entry and processing.

Even if it was in a not very targeted form and rather slowly, the breakthrough of word processing did eventually happen. And only when the word processor was actually there, did it become apparent what latent need it filled. As the concept became clearer, its possibilities came more into view, and more functions could be developed and further refined. In this way the word processor was finally able to move—however slowly—away from the typewriter. It developed from an application which rigidly distinguished between the entry of text and its editing into an application in which text entry, storage, editing and all sorts of further operations were fully integrated. Today's word processing application is meant to serve both the thinking writer and the office clerk. Ultimately perhaps it never really succeeded in being all to everyone. Certainly the word processor is more suitable for office use than to assist in the creative thought process. As tools in the thinking process new applications more similar to what Engelbart had in mind have since been developed, such as outliners and mindmappers.

Given that the preconditions for the capability of text processing were already theoretically present around 1950, the development of word processing as a functional concept was actually quite slow. It is not difficult to identify reasons for this slowness. Without doubt the main reason was insufficient demand for storage of text for reuse on the social side. Apart from the very simplest form (the creation of a 'correct text') absence of any demand for manipulation of the entered text. The ideas of Engelbart and Nelson were too idealistic and sophisticated to lend themselves to mainstream applications. Employing computers for the tasks that typewriters were used for in the 1950s and 1960s—simple

---

*International Ludwig Wittgenstein-Symposium in Kirchberg am Wechsel, Austria 2007*, ed. by A. Pichler & H. Hrachovec, Publications of the Austrian Ludwig Wittgenstein Society, New Series, vol 6., Frankfurt a. Main, 2008, pp. 185-211.

[27] It had been previsaged by Georges Bafour, André Blanchard and François Raymond (BBR) in France in 1954 (L.W. Wallis, *A Concise Chronology of Typesetting Developments, 1886-1986*, London, 1988; 2nd edn, Upton upon Severn, 1991, p. 27).

office use for correspondence and reports, or the small market for specialist purposes such as for typesetting machines—was simply too expensive. In addition, other priorities had prevailed for the development and deployment of computers during WWII. Computers were a product of the military-scientific complex; it was only there and in the largest civilian companies that there was enough money for the acquisition and development of advanced equipment and applications. In the military, sophisticated ballistical calculations swallowed up the money and the best minds. In offices, manipulation of text at the elementary level of the average secretary was simply not at issue. This situation did not change substantially until after the mass-produced microprocessor became available in the mid seventies.

It is also in this light that the famous prediction by Douglas Hartree, professor of mathematics at Cambridge in the 1950s, should be regarded: that the world would never require more than five computers.[28] Such predictions were obviously based on a calculation of the need for computing power for such things as were then being done with computers. These were mathematical calculations, preferably of things which, once calculated, would remain ever valid and useful, such as the census results or nautical tables (compare the tables that Babbage's Differential Engine could produce). Such calculations could neither be too short, or it wasn't worth the investment of writing the program, nor too long, or the error-prone computers of the time would not run for long enough to finish the calculations. Had Hartree been able to consider the cheap microprocessor in combination with the possibilities of text processing he would no doubt have come to very different conclusions.

In sum, that a clearly existing, if latent, need for the functionality of word processing was not being translated into product development resulted from two factors, which were locked in a vicious circle. First, computers and programmers were too scarce and expensive to allow sufficient demand for an application like a word processor to arise. Secondly, there was the inability to imagine the functionality of an word processor, so that insufficient demand could arise for the devices on which those applications could run.

> In the mid-seventies, several years before the Apple II first exploded onto the marketplace, an Intel engineer called a meeting of the company's board of directors to make an impassioned case for building a personal computer. He rolled out his vision of a future where consumers bought digital machines for their homes the way they currently bought televisions, stereos, and vacuum cleaners. The fact that Intel already possessed the technology—the chips, the integrated circuitry, the power supply—to make a machine for less than ten thousand dollars made the case a particularly compelling one, even though the behemoth mainframes of the day regularly sold for hundreds of thousands of dollars. But the board wanted an answer to a question that seems self-evident to us today: what were people going to do with these personal

[28] 'I went to see Professor Douglas Hartree, who had built the first differential analyzers in England and had more experience in using these very specialized computers than anyone else. He told me that, in his opinion, all the calculations that would ever be needed in this country could be done on the three digital computers which were then being built—one in Cambridge, one in Teddington, and one in Manchester. No one else, he said, would ever need machines of their own, or would be able to afford to buy them.' (Lord Bowden, 'The Language of Computers', *American Scientist* 58 (1970) pp. 43–53, on p. 43.

computers? Amazingly enough, the engineer didn't have a satisfactory answer: his most compelling scenario involved filing electronic versions of cooking recipes. Of all the eventual hightech applications devised for the personal computer, all those spreadsheets and word processors and video games, the best he could come up with was a digital version of Mom's tuna casserole. It was like inventing the wheel and then immediately demonstrating what a wonderful doorstop it made.[29]

Such an attitude stems from thinking patterns that are determined by familiarity with existing technologies and practices, such as in this case the typewriter, and system cards in a card index. Using the technology of the typewriter the emphasis was on capturing and transferring an existing text as correctly as possible, not on the integration of thought and reflection in the writing process. If one had been able to imagine the functionality and had made it available (cheaply enough), it would certainly have generated a demand. (In the case of the mobile phone, there was also no question of an explicitly existing demand; the mere fact that the functionality became available apparently woke a dormant need for a device that allows people to talk to each other at all times of the day and night.)

## 4. Document management

Ted Nelson already made an appearance in his capacity as one of the most visionary developers of the concept of hypertext (and inventor of the term). In many respects Nelson was inspired by Vannevar Bush (in his *Literary Machines* of 1981 Nelson reprinted Bush's complete essay 'As We May Think'). It is Vannevar Bush who has gone down in history as the 'onlie begetter' of the principle of hypertext. Despite many similarities one big difference between Bush and Nelson was that where Nelson was mainly interested in the creative writing process, Bush focused more on document management. In his workstation for the scientist, called the 'Memory Expander', or simply 'Memex', he had designed an ingenious system for storing and linking existing pieces of information (and annotations about them). In 'As We May Think', the famous article in which he introduced his Memex in 1945[30] Bush suggested that, now that the war was over, science was facing a whole new challenge. That challenge was the task of making the expanding volume of human knowledge more accessible and more manageable. Instead of man's physical force it was time to strengthen the power of the human brain. Man had to gain more control over the knowledge he had accumulated in the course of the centuries. To that end, Bush devised an organising principle that was based on associative links instead of the usual alphabetical and systematical referencing systems. Such a principle, according to Bush, did more justice to the way the human brain works. In the microfilmed books, articles, notes and correspondence contained in his Memex the investigator could leave 'thinking trails' that

---

[29] Steven Johnson, *Interface Culture*, p. 148. With the knowledge of hindsight we would perhaps not place recipes at the top of the list of promising applications, but as David Weinberger writes in *Everything Is Miscellaneous: The Power of the New Digital Disorder*, New York, 2008, pp. 44-45, the computer can certainly reinvigorate the genre of the cookbook.

[30] Vannevar Bush, 'As We May Think', *Atlantic Monthly*, July 1945, pp. 101-8 (also at http://www.theatlantic.com/magazine/archive/1969/12/as-we-may-think/3881/)

linked relevant sources by means of assigned codes. These thinking trails remained available to the user at all times and thus formed an ‘intimate supplement to his memory’.

> It consists of a desk, and while it can presumably be operated from a distance, it is primarily the piece of furniture at which he works. On the top are slanting translucent screens, on which material can be projected for convenient reading. There is a keyboard, and sets of buttons and levers. Otherwise it looks like an ordinary desk.
>
> In one end is the stored material. The matter of bulk is well taken care of by improved microfilm. Only a small part of the interior of the memex is devoted to storage, the rest to mechanism. Yet if the user inserted 5000 pages of material a day it would take him hundreds of years to fill the repository, so he can be profligate and enter material freely.
>
> Most of the memex contents are purchased on microfilm ready for insertion. Books of all sorts, pictures, current periodicals, newspapers, are thus obtained and dropped into place. Business correspondence takes the same path. And there is provision for direct entry. On the top of the memex is a transparent platen. On this are placed longhand notes, photographs, memoranda, all sort of things. When one is in place, the depression of a lever causes it to be photographed onto the next blank space in a section of the memex film, dry photography being employed.
>
> There is, of course, provision for consultation of the record by the usual scheme of indexing. If the user wishes to consult a certain book, he taps its code on the keyboard, and the title page of the book promptly appears before him, projected onto one of his viewing positions…
>
> It affords an immediate step … to associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another. This is the essential feature of the memex. The process of tying two items together is the important thing.
>
> When the user is building a trail, he names it, inserts the name in his code book, and taps it out on his keyboard. Before him are the two items to be joined, projected onto adjacent viewing positions. At the bottom of each there are a number of blank code spaces, and a pointer is set to indicate one of these on each item. The user taps a single key, and the items are permanently joined…
>
> Thereafter, at any time, when one of these items is in view, the other can be instantly recalled merely by tapping a button below the corresponding code space.

These thoughts of Bush’s must have spoken to the imagination of his contemporaries. The system that Bush had in mind not only gave the user access to all relevant scientific information. He could also organise that information in a way that was tailored to him personally.

The creation of personal thinking trails through the linking of documents is central to Bush’s concept. However, it is somewhat misleading to base the role of Bush—and others—in the creation of hypertext entirely on that. Practically speaking the notion of hypertext quite changed in the course of time. To Bush the focus had been on the storage and consumption (organising and providing access) of knowledge produced by others, less on the production of new knowledge (although that was certainly facilitated by the Memex), and not at all on distribution (sharing new knowledge). The restrictions are obvious. For example, the operator of the Memex is not connected to other users; access to the stored data is limited to one person. While microfilm as a technology has now all but become obsolete, at the time it held a great promise of miniaturisation. Yet what is—again—striking is that he makes no mention of the computer at all, while as director of the Office of Scientific Research and Development during the war he had witnessed the feverish activity that had gone into its development. It confirms our earlier observation that at that time the worlds of text and the computer were still far apart.

One might argue that it is unfair to express such criticism with the benefit of hindsight. But for example the Belgian Paul Otlet had conceived the idea of universal access to centrally stored knowledge as early as the 1930s. A network of 'electric telescopes' would enable users to browse millions of linked documents, images, and fragments of sound and moving images, to send each other messages, and even to meet in virtual communities.[31] That Otlet did not manage to secure for himself a more prominent place in the history of the Internet and the World Wide Web is tragic in light of his visionary ideas. But it is perhaps indicative of the fact that the focus of information management was already beginning to migrate to the other side of the Atlantic Ocean.

Despite the fact that Bush was using the technologies of his time and did not think it a problem that the material in the Memex was tailored to the wishes and needs of a single scientist, it is clear that his views on the individualisation of the organisation of information played a very important role in the development of the textual world in which we now live. Bush recognised the need for a different way of organising and dealing with textual knowledge. What we also have largely to thank him for is the vision that knowledge may consist of discrete chunks of text on the user's system, and that those chunks files may in some way be connected with one another.

However obvious it may now seem, again it took time before the concept of 'files' on the computer representing such chunks of information belonging together was developed. Texts or even fragments of text were initially simply placed *seriatim* on the storage medium, with codes indicating the beginning and end of the corresponding parts. The importance of files has only recently started to recede again. As a method to control access to information on the hard disk it is being overtaken by fulltext retrieval. Since PCs have become powerful enough, the hard disk (and every other storage medium connected to the computer) is being permanently indexed and text can be retrieved by keying in a distinctive word, regardless of where the file is physically or logically located. Ironically enough, we might be said in a certain way to be returning to the old situation. Before long the distinction between documents and the 'information space' to which individual documents belong will probably no longer be significant. The PC's hard disk is turning into a miniature Internet, which may be accessed through the services of a miniature Google. In fact, with the increasing popularity of web-based applications and disk storage soon the distinction between a local hard disk and the public Internet is likely to fade still further.

---

[31] See Armand Mattelart, *The Information Society: An Introduction*, London and Thousand Oaks, 2003, pp. 42-43, and Françoise Levie, *L'Homme qui voulait classer le monde: Paul Otlet et le Mundaneum*, Bruxelles, 2006.

5. GUI en WYSIWYG

In 1979 a delegation of Steve Jobs' Apple Computer company was offered a tour of Xerox' Palo Alto Research Center (PARC). Here they were shown a remarkable human–computer interface that experimented with a wholly new way of giving instructions to the computer. Opinions differ as to Apple's indebtedness to Xerox for the main concepts used in their revolutionary operating system.[32] Whatever the case may have been, Apple ended up hiring quite a number of ex-Xerox staff to build the Apple Lisa of 1981. The Lisa sported a Graphical User Interface (GUI) with a Windows, Mouse and Pull-down Menus (WIMP) environment. It closely resembled what Xerox had been experimenting with (but had not seen fit to develop commercially), based on original ideas going back to the early 1970s, when Xerox had the ambition to challenge the dominance of IBM.[33] The Lisa lay the foundation for the Apple Macintosh with its advanced graphic capabilities and its legendary user-friendliness. Through the metaphor of desktop, folders and files a graphic façade was built which screened the ordinary user from the raw reality at system level.[34] This trend to shield the user by creating a black box around complex technology is also evident more generally in the design and further development of the World Wide Web.

As early as the 1960s Douglas Engelbart had been experimenting with various input devices to aid human–machine interaction. Among his many achievements was the invention of the mouse, patented in 1970. Engelbart's efforts were primarily focused on scientific users. Apple's GUI interface, however, made the computer more accessible also to laymen, who were not necessarily waiting to be induced into the esoteric world of the computer geek. By the 1980s it had become clear that with the arrival of the microcomputer and personal computing, interfacing was no longer just a concern of professional users, but had become a major general issue. When the computer came increasingly to be used by people who were not programming experts and had little interest in the technological side of computing, the need for a more intuitive user interface became pressing. That the Association for Computing Machinery (ACM) organised its first Computer–Human Interaction conference in 1982, one year after the presentation of Apple's Lisa, was therefore not a coincidence.[35] In the WIMP environment the entry of often very cryptic letter combinations on the command line was replaced by foldout menus and the like worded in ordinary language, even though the language was wrapped in

---

[32] Some say the word 'theft' is not too strong. However, John Seely Brown and Paul Duguid (*The Social Life of Information*, Boston, 2000, pp. 159-61) stress that what Xerox showed Apple were far from marketable technologies. The concept of the GUI needed 'several advances in the hardware and software design'. It took Apple all of those five years from 1979 to 1984 to get it right.
[33] Brown and Duguid, *The Social Life of Information*, pp. 150-51.
[34] Steven Johnson, *Interface Culture*, p. 15.
[35] Again, not coincidentally, the 1980s saw a tremendous outpour of books on interfacing, such as Ben Shneiderman's *Software Psychology: Human Factors in Computer and Information Systems*, Boston, 1980 and *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Reading, MA, 1987; Donald Norman, *The Psychology of Everyday Things*, New York, 1988; P. Ehn, *Work-oriented Design of Computer Artifacts,* Stockholm, 1988 (see also Christine L. Borgman, *From Gutenberg to the Global Information Infrastructure: Access to Information in the Networked World*, Cambridge, Mass and London, 2000, pp. 118-19).

graphical elements. However, it is not easy to make icons unambiguous, and to this day they are almost always accompanied by textual explanations, frequently in the form of a 'mouse-over'.[36] Communicating in symbols other than linguistic ones is not easy for humans. That goes for social communication, but also for human–machine communication. The GUI interface reinforced the textual foundation on which the whole human–machine interaction had come to rest ever since the introduction of higher-level machine languages.

The WIMP and GUI interface led toward the What-You-See-Is-What-You-Get (or WYSIWYG) possibilities of DeskTop Publishing (DTP). The classic example was Aldus' PageMaker for the Apple Macintosh, introduced in 1984.[37] What WYSIWYG word processors and layout programs offered, was an improvement of the typographical capabilities of the computer. This resulted in a better connection to the textual environment outside the computer, which is based on centuries of typographic conditioning. It made it possible to use the computer for laying out text graphically for reproduction in print.

DTP, in other words, was again a product of our mindset as *homo typographicus*, or typographic beings. It was designed to produce conventional print. All popular reproduction techniques of the 1960s and '70s, such as offset lithography, electronic stencils and photocopying, used the same method for origination of the printing forme (i.e., to produce camera-ready copy): cut-and-paste. The cut-and-paste method derives its name from the way text set in galleys by means of phototypesetting was cut up to page or column lengths and pasted on to imposition sheets. These imposition sheets were photographically transferred to printing plates (or electronic stencils) for multiplication. In this method, before the text could be typeset, the specifications had to be decided by a designer, who relied on expert knowledge and prior experience to envisage the visual effect of any typographic instructions. Any trial-and-error approach to design was forbiddingly expensive. Corrections to the typeset text required further keyboarding by the typesetter, the generation of fresh galleys, and a fresh round of cutting and pasting. The cost of correction thus formed a significant expense in print production.

The WIMP and GUI environment of the Apple Macintosh provided the environment in which the cut-and-paste process could be performed digitally by the computer. In this way DTP brought the full functionality of computerised typesetting—and more—to the personal computer. Instead of typesetting the text first in galley lengths for physical cutting and pasting, computer layout programs such as PageMaker allowed the layout of the text *prior to* typesetting, in exactly the same way as word processing had allowed editing and correction prior to printing out a text. The result of the typographic 'encoding' was in these circumstances no longer a matter of typographic knowledge and imagination, as it had been in the pre-DTP days, but became instantly visible.

Typing, whether or not in the shape of word processing, in offices was and remained a clerical job, performed by secretaries and data entry typists. Similarly in the typesetting industry composing type for printing was a heavily unionised trade and so a jealously

---

[36] Edward Tenner, *Why Things Bite Back:Technology and the Revenge of Unintended Consequences*, New York, 1996, pp. 194-96.
[37] The earliest WYSIWYG program had been Bravo, produced at Xerox PARC in 1974.

guarded oligopoly. But when miniaturisation and microchips ushered in the era of 'personal computing' this changed irrevocably. In combination with Adobe's PostScript page description language of 1969, which enabled the WYSIWYG layout to be output to a laserprinter or phototypesetting machine, PageMaker caused the DTP revolution of the mid and late 1980s. Page layout programs took the graphic design industry by storm, and DTP democratised the origination stage of print production.

The greater user-friendliness of the GUI/WIMP interface began to shift labour away from the trained specialist worker, and eventually caused a shift in control over the entire production process from the editorial office to the author. Authors, who had already begun to submit their word processed work on disk, could now become responsible—voluntarily or by publishers' request—for providing finished camera-ready copy to the publisher. Especially in the case of scholarly publishing, where direct monetary rewards for the author are rare anyway, the author or editor was often placed in charge of the entire production process leading up to actual printing.

The advent of WYSIWYG computing and easy-to-use DTP applications in turn accelerated and completed the shift from letterset printing to offset lithography. This had far-reaching consequences for the 'democratisation' of print production. It succeeded in breaking the hold of specialist workers and their unions on print production, completing the process of liberalisation begun with the introduction of offset lithography in combination with various crude cut-and-paste techniques. It was a powerful combination, as the student-led political protest movements of the 1960s clamouring for the democratisation of the means of production had already discovered. While a comparison between this minor twentieth-century print revolution and Gutenberg's invention that created the original Order of the Book might seem far-fetched, it is highly unlikely that the 1960s and 1970s could have become the era of political protest, mass democratisation, 'underground' culture and popular revolution that they were without the proliferation of cheap and accessible print enabled by offset printing.

For a decade the extreme user-friendliness of its GUI/WIMP interface gave Apple, at least in the graphic industry niche, an unchallenged competitive edge over Microsoft with its command-line operating system, MS DOS. Observing the success of the Macintosh, in 1983 Microsoft announced its own version of a GUI/WIMP environment, which was to be called Windows. Its original release in 1985 was no great success, however, and it was not until version 3.0 in 1990 that it gained significant third-party support. From that moment, however, it proceeded to take the world by storm. For better or for worse, and whoever ought to be credited with the honour, Microsoft's development of Windows meant the end of the command line interface for the average computer user, and it firmly established the GUI/WIMP environment and WYSIWYG as the way forward.

This represented a major victory. *Homo typographicus* had an intuitive tendency to transfer an irredeemably typographical view of text to the digital environment as much as possible. The word processor for example, has progressed from codes that were visible to the user via codes that could be made visible on request (as in WordPerfect's famous 'underwater screen'), to totally invisible and inaccessible codes, as in most word processing applications today. While continuing to perform the same function, the visible formatting markup of old has now become invisible; at the same time the effect of that formatting is now instantly visible where once it could only be imagined. The desire of *homo typographicus* to look at the typographical surface of text rather than its underlying 'raw' structure contrasts with the more structured view taken by the specialised professionals. This applies to both the human–computer interaction, and to the manner in which they regard text. It is worth noting that, ironically, precisely by making the computer into a better tool to continue existing analogue practices, *Homo typographicus* actually severely curtailed the computer's potential to foster greater awareness of the logical structure of text. The WIMP / GUI environment has not only exiled the commands and codes that structure the typographical text to a place behind the scenes, but also that logical view of text in general.

## 6. Markup

There is no doubt that the GUI has made the computer incomparably more user friendly. However, those who dealt with books professionally had other priorities than user-friendliness. The sheer volume of documents being processed, notably in the publishing world, stimulated a more structured approach to text. The exigencies of scale pointed inexorably in the direction of automation, which placed very different demands on the computer. For those who handled text professionally the steep learning curve in acquiring the necessary knowledge and skills for a more structured approach to text constituted no obstacle. Even though typography as yet remained the ultimate goal (for it would still take some time for the digital transmission of text via the Internet to come off the ground), as early as the 1960s the insight had begun to take hold that a text's content and its typographic appearance could be separated from each other in a digital environment, and that this was, moreover, a meaningful distinction to make in the editing and production process. This led to the recognition that all text had an underlying structural pattern, regardless whether this was made explicit through computer encoding or remained implicit in the text's typography.

Chapter 2 showed how documents can be classified into categories, or 'types', on the basis that they contain the same structural components, regardless of how they may look typographically. Publishing companies and typesetting establishments were well used to employing codes as a short description of the typographic format for chunks of text. In the case of an article for a scientific journal there might be codes to identify such indispensible elements as author, title, synopsis, and the text itself, consisting of, for example, paragraphs, citations, illustrations, footnotes, and a bibliography. From this existing encoding practice aimed at typographic form it was not such a very big step to apply encoding to structure instead of form. The recognition of the structural nature of document types was based on the understanding that a text's structure was not dependent on its (accidental) layout, but that structure is already implicitly present in every text, being as it were brought to the light by thetypographic form. This led to the development of the Standard Generalised Markup Language (SGML), which was accepted as an ISO standard in 1986, and later the eXtensible Markup Language (XML) already discussed in Chapter 2, which was accepted as a standard in 1998. Introducing this markup in the editing and production process of large publishing houses resulted in significant savings. This applied not just to the production in print of the publications themselves, but also for streamlining the entire editorial workflow. In the SGML/XML encoding could also be included all kinds of metadata, such as the authors' contact details, the date of delivery, particulars about the editing and peer-review process, and so on.

The principle of markup, with its emphasis on structure instead of typography was an approach to text in line with the logical approach to text that was encouraged by the logical nature of the computer. An additional advantage was that markup opened up far-reaching possibilities for all sorts of other advanced text processing. Initially this concerned especially publishing houses, where there was an immediate economic need for them, and some scholarly applications. But ultimately markup proved to be interesting also for a wider audience. The main impetus for the popularisation of markup came from its deployment on the Internet, with the creation by Tim Berners-Lee of the HyperText Markup Language (HTML) and the World Wide Web, followed by the first graphical Web browser, Mosaic, developed at the American National Center for Supercomputing Applications. HTML is a simplified application of SGML.

It is not inconceivable that if word processing under MS-DOS and the command line had remained in existence longer, it would have resulted in a wider acceptance of markup as an alternative way to regard text.[38] The graphic developments in computer text processing, as in human-computer interaction, which responded to the wishes of *homo typographicus*, have thus led to parallel worlds of digital textuality that only partly overlap.

## 7. Networks and standards

This chapter has so far been mainly concerned with the production stage of digital text forms. But the advent of text to the computer has culminated in the Internet and the World Wide Web. This has resulted in major changes also in the distribution and eventually also consumption of texts. The pre-digital distribution of text in other than paper form has a long history, which warrants at least a cursory glance. The telegraph was the first method that enabled rapid remote communication without the visual limitations of smoke and semaphores. Storage on punched tape made it possible to send the same text to multiple recipients. The telex was an improvement on the teletype with its fixed lines, and made it possible, as with a telephone, for an individual connection to be made with any other subscriber connected to the system. It was mainly a system for the military, government and industry. Although everyone could make use of it through the services of post and telegraph offices, there was no direct public access to the system.

The instant distribution of text across the globe was thus admittedly not new, yet connecting computers in networks was a crucial new development. As Manuel Castells has phrased it, 'the Internet was born at the unlikely intersection of big science, military research, and libertine culture'.[39] The first networks arose from the need to share scarce

[38] As it is, the functionality of markup is approximated most in the use of styles in a word processor.
[39] Manuel Castells, *The Internet Galaxy*, OUP, 2001, p. 17.

and therefore expensive computer time. Not long afterwards, the Arpanet (the Advanced Research Projects Agency Network; later, with the addition of 'Defense', Darpanet) was founded.[40] By means of a network terminal application it was possible for clients to give instructions to the host computer. Soon (1971) Arpanet also gained a function for human communication, through e-mail—an alternative to the phone. The 'inter'-net originated when the existing networks such as Arpanet, PRNET and SATNET were linked in a new 'network of networks'. The importance of this military origin for the technological characteristics of the Internet will become clearer in the next chapter.

The 'grassroots tradition of computer networking'[41] has had further major implications. The internet is based on three principles: 'a decentralised network structure; distributed computer power throughout the nodes of the network; and redundancy of functions in the network to minimise the risk of disconnection. These features embodied the key answer to military needs for survivability of the system: flexibility, absence of a command center, and maximum autonomy of each node.'[42] The packet switching architecture, in which all messages are broken up into small packets which can reach their destination through any route before being reassembled at their destination, explains how the network can be as reliable as it is. These design principles still define much of the nature of the medium as it has developed since.

In the 1970s and 1980s use of the budding internet gradually widened from the initial select group of scientists and military users. It wasn't till the birth of the World Wide Web in 1991 that the real expansion of the computer-in-a-network began. This gave hypertext the chance to grow from an authoring system (first hinted by Vannevar Bush and invented by Theodore Nelson) to a publication format. The World Wide Web as Tim Berners-Lee implemented it in HTML offered much of the functionality that Bush had already described (with the notable exception of making annotations to sources), but there was less room for Nelson's more advanced ideas about authoring and collaboration. It is only authors and/or publishers who manage and control the links to pages that they publish, and HTML offers no straightforward two-directional traffic between author and reader. Also, the unit of information is much larger than the 'lexias' (the smallest meaningful elements in a text eligible for reuse) anticipated by Nelson. With the development of the Internet and then the World Wide Web digital text processing expanded further. In addition to digital composition, editing and production there now also arose the possibility of distribution in digital form and hence that of digital consumption. This meant that it was no longer necessary for the digital process to culminate in an analogue product by printing out files and messages.

When transmission over the network in one form or another (for the exchange of programs or data) gained in importance, standardisation obviously became imperative. In the era of letterpress, printing standards differed widely, with different national systems for measuring type size and different type heights. This is not to say that no one was interested in standardisation. The DIN (Deutsche Industrie Norm) paper sizes like A4, for example, have caught on fairly widely (though by no means universally: they are not used

---

[40] Castells, *The Internet Galaxy*, p. 10.
[41] Castells, *The Internet Galaxy*, p. 12.
[42] Castells, *The Internet Galaxy*, p. 17.

in the U.S.). It is simply that in the physical world of print standards don't play a significant role, and so standardisation was simply never achieved beyond some few limited domains. In a computer network designed to exchange information it is no longer feasible—let alone desirable—to allow local, or even national, standards to flourish in the way they proliferated in the print era. Just like the advent of international long-distance trains spelled the end of the patchwork of national railway gauges in Europe the internet too requires full standards compliance to ensure smooth traffic.

Standards for efficient and effective communications between computers are needed at different levels. For character representation the 1963 American Standard Association's ASCII (American Standard Code for Information Interchange) became the standard 7-bit code for data transfer. It is still used on most personal computers in the Western world.[43] In the ASCII table each character, number or symbol equals one byte of information. For example: 01000001 represents A, 01100001 represents a, and 00110011 represents 3. ISO Latin 1 (ISO standard 8859:1) is an 8-bit set which provides a standard encoding for accented characters in Latin scripts. Importantly, ASCII as the lowest common denominator of computer character encoding remains in use for markup in SGML and XML, with HTML as their ubiquitous and most visible application.

By the late 1980s disk space and memory constraints had relaxed sufficiently for a 16-bit encoding system, called Unicode, to become a feasible alternative. In 1991 version 1.0 of Unicode was launched, and it has been under development since. By using two or even more bytes per character it aims to represent a much larger proportion of the world's characters in a single system. Though it still hasn't covered them all, the Unicode standard represents a major step forward, and despite its slow development it has attracted widespread support. It has been implemented in many technologies, including XML, web browsers, and most operating systems.

Apart from the display of text, there are also standards for its transmission (the TCP/IP protocols and their predecessors) and application protocols, of which email and file transfer (FTP) are the most important. When the World Wide Web was implemented the standards for web pages were added: first HTML as originally designed by Tim Berners-Lee and derived from SGML; later also XHTML and XML, with associated stylesheet languages, such as CSS, and JavaScript. Although some 'de facto standards' still exist (that is to say, conventions that are regarded as standard purely on the basis of their wide dissemination), the trend is clearly in the direction of open standards. It is by dint of standardisation that many so-called Web 2.0 applications (such as mashups—the combination of data from different sources into a new application) are possible.

Standards also play a role, albeit a less crucial one, in document formats, for example, of word processors. As in the case of text representation, standardisation of document formats was not an issue as long as computers remained expensive and independently operating appliances. When the microchip led to the mass production of pcs a host of de facto standards came into being, which were created by accidental market share. A well-known example is how the current hegemony of Microsoft Word followed the decline of WordPerfect, which did not make a successful transition to the GUI era. When

[43] Many mainframes still use Extended Binary Coded Decimal Interchange Code (EBCDIC), descended from punched cards, and devised by IBM at the same time.

exchange of information became more important, and *a fortiori* when the network started to experience strong growth, the usefulness of standards for document formats and markup became gradually more widely recognised. In addition it became clear that de facto standards had their limitations. The users of commercial file formats depend on the supplier (they must have a license themselves, but so must everyone who wants to be able to open the file), and the company may go bankrupt, ending support. Open standards can prevent such ‘lock in’ and are above all more future proof. The expectation that standards, whether open or de facto, are eternal has appeared to be an illusion. The only feasible solution is to devise standards which don’t just make it possible but also easy to be replaced in the future by new standards. The ideal standards are publicly available, transparent, well documented and as far as possible platform independent.

## 8. Conclusions

This chapter followed the development of computers as independent mechanical calculating machines to digital computers linked in a network and functioning as a textual medium. The social use of the computer has developed too, from individual usage patterns to more social and communicative ones. It has been a development of hardware, but also of software. The history of the representation of text and the use—and understanding—of the role of standards is a good example of the extent of the transformation involved.

The development of the digital textual medium has been a hesitant process, with a great deal of trial and error, along a winding route. The processing of text initially represented an attempt to transfer a thoroughly familiar typographical world view wholesale to this new medium. This tendency to regard the new in terms of the old is a very human one. Itcould also be observed in the case of the introduction of printing with individual lead types. The first printed books could hardly be distinguished from the manuscripts that they replaced. However, neither is a case of conscious imitation. Rather, it is a matter of involuntary following familiar conventions and continuing along a known road. It is inevitable that the possibilities that the new medium offers are not always fully recognised at the time, even though they may look obvious in retrospect. There are always people with enough imagination and originality to sketch that kind of potential, but if their thinking insufficiently connects with what is usual at the time, their ideas will not be recognised.

Though I have deliberately painted an image of hesitation, trial and error and a meandering path this may be somewhat misleading in at least one respect. For it would be wrong to believe that it took *a long time* before the computer became a major force in text processing. Compared with the slowness with which humans learned to read and write, and even with the time needed for the triumph of the printing press, the transformation of the computer as a number cruncher to a digital medium happened at a break-neck speed. Despite the trial-and-error fashion in which it happened, a mere half a lifetime saw the creation of the digital computer, its appropriation for an extremely efficient process of writing, editing and the production of printed matter, *and* its transformation into an entirely new worldwide medium in which text, images and sound can be mixed at will and transferred across the globe instantly.

Nor was that even all. The Internet—and later the World Wide Web—is just one of the many things that the computer has been deployed to create. The printing press could only do one thing: multiply text. The computer, by contrast, is the most versatile machine ever devised by humans. Even just with regard to medial applications, its possibilities are limitless. Besides text, it now also processes still images, sound and moving images digitally. The Internet may be regarded as the 'ultimate' medium. Unlike any other medium, the computer-in-a-network not only connects all links in the communication chain of production, distribution and consumption, but also other modalities and so allows the ultimate convergence of all traditional mediums. It unites virtually all properties of all existing mediums.[44]

Within those medial possibilities this book covers text only—although text is, as I have argued, a crucial component in the mix of modalities. The processing of text was probably the first, and certainly the most important, symbol-manipulatory uses of the computer that was not aimed at a numerical outcome. The great significance of being able to use the computer as word processor was that it made text into something fluid and pliable. Electronic digitality gives text (like all digital modalities) the property of virtuality (whose significance will be the subject of the next chapter). Initially the computer was no more than a 'prosthesis': an aid for the creation of analogue text. Word processing created a digital production environment for the creation and storage of text. This might be called the level of 'digitality per se': the existence of textual information in an electronic digital format. It was the creation of networks of computers, combined with standards of all kinds, for example for markup, that made the computer into a medium—a medium that far exceeds the production of the printing press. The network makes use of electronic digitality to multiply and distribute the flexible text. In the history of textual transmission the computer-in-a-network adds to the recording of text (in characters) and its multiplication (in print) the dimension of its distribution. Calling the digital text a medium is saying something very different than calling writing or the printing press a medium, even if all of these mediums comply with the definition that I gave at the end of Chapter 2: 'A medium is a construct consisting of a tool or technology with its (explicit) technical protocols and any implicit social protocols with the purpose to communicate information expressed in one or more of the modalities of still text; images; sound; and moving images over time and/or space.'

What is particularly noticeable in the development of the digital textual medium is its absolute interdependence with the development of the computer in a more general sense. In other words, it is not as if the hardware and software required for the medial function were developed wholly independently from the hardware and software needed for other applications. The processing of text served an end in itself but was also a means to control the computer more efficiently. The Internet got a medial function through which humans could communicate with other humans, but it arose from the need to be able to communicate remotely with the computer itself—the few large mainframes that once used to be all the world knew.

---

[44] Although the broadcast opportunities are still limited by the current bandwidth.

The inseparableness of the development of the Universal Machine and the digital textual medium is illustrated well by the dualist—or hybrid—view of text that persists today. On one hand it has proved possible to refine the typographical rendition of text on the computer to a high standard of sophistication. That does justice to the fact that after so many centuries of the Order of the Book *homo typographicus* has come to regard the use of typography as an integral part of the writing process. *Homo typographicus* uses typography to structure the text while writing. WYSIWYG in word processors is an aid to such structuring, and the PDF format makes it possible to preserve the subtlest typographic effects during the transmission of the text. On the other hand, the computer has served to emphasise the logical or structural form of text as it is reflected in markup as a Ordered Hierarchy of Content Objects (see Chapter 2). This hybridity is nicely illustrated by Tim Berners Lee's HyperText Markup Language (HTML)—that crucial step in the creation of the computer-as-medium. HTML eagerly makes use of achievements from our typographic history (such as the <i> and <b> elements to render purely typographical conventions like italic and bold) but no less of new achievements that only come into their own in a digital context (such as the hyperlink or the addition of metadata to a document).

Typography has a long track record as an aid in structuring text. Slowly the lesson is being learned that although typography may have become a second nature to humans, the computer is severely handicapped in a typographical environment. This does not of course mean that typography would have no future in the digital environment. It does mean that humans will need to give more attention to the logical form of text, even if that is much less intuitive. For the logical form of text is better suited to the nature of the digital technology and will enable the computer to process the text in many further ways than a typographic rendition. What exactly that nature is, is the question the next chapter will deal with, through an analysis of the intended and unintended primary and secondary properties of the new medium.